Structural analysis for the diagnosis of distributed systems

Carlos Gustavo Perez Zuniga

Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par : *l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*
Discipline ou spécialité : *Doctorat de Automatique et Informatique*

Présentée et soutenue par : *Carlos Gustavo Pérez Zuñiga*
Le *21/08/2017*

Titre : *Analyse Structurelle pour le Diagnostic des Systèmes Distribués*

JURY

Vincent COCQUEMPOT, Professeur des Universités, Université de Lille.
Addison RIOS-BOLIVAR, Full Professor, Los Andes University.
Antonio MORAN CARDENAS, Full Professor, PUCP.
Louise TRAVÉ-MASSUYÈS, Directeur de recherche, LAAS-CNRS.
Elodie CHANTHERY, Maître de conférences, INSA Toulouse.
Javier SOTOMAYOR MORIANO, Full Professor, PUCP.

Ecole doctorale : *EDSYS : Automatique 4200046*
Unité de recherche : *Laboratoire d'Analyse et d'Architecture des Systèmes LAAS - CNRS*
Directeurs de Thèse : *Louise TRAVÉ-MASSUYÈS, Elodie CHANTHERY et Javier SOTOMAYOR M.*
Rapporteurs : *Vincent COCQUEMPOT et Addison RIOS-BOLIVAR*

Résumé

Les récents développements des systèmes technologiques ont mené à une complexification des comportements des systèmes actuels. Une solution pour gérer cette complexité croissante consiste à les considérer comme un ensemble de sous-systèmes hétérogènes et à développer des techniques distribuées pour les contrôler et les gérer. Cette solution soulève plusieurs problèmes. Tout d'abord, l'augmentation de la taille et du nombre de composants entraîne immanquablement l'augmentation du nombre de fautes qui peuvent conduire le système dans un état de défaillance critique. De fait, parmi les fonctions opérationnelles, les tâches de détection et d'isolation des fautes (Fault Detection and Isolation ou FDI), de maintenance et de réparation sont devenues prédominantes et elles influent considérablement sur le coût total des produits finaux.

Cette thèse porte sur la détection et l'isolation de fautes. Parmi les différentes méthodes pour générer des tests de diagnostic utilisant la redondance analytique, cette thèse adopte l'approche par espace de parité qui utilise les relations de redondance analytique (RRA). Étant donné un modèle du système sous la forme d'un ensemble d'équations différentielles, les RRA sont des relations obtenues à partir du modèle en éliminant les variables non mesurées. Ceci peut être effectué dans un cadre analytique en utilisant la théorie de l'élimination. Une autre solution consiste à utiliser l'analyse structurelle. L'analyse structurelle est basée sur une abstraction du modèle qui ne conserve que les liens entre variables et équations. Malgré son apparente simplicité, l'analyse structurelle fournit un ensemble d'outils puissants, s'appuyant sur la théorie des graphes, pour analyser et inférer des informations sur le système. Par ailleurs, elle a l'avantage de s'appliquer indifféremment sur les systèmes linéaires ou non linéaires.

L'objectif de cette thèse est de développer des techniques efficaces basées sur l'analyse structurelle pour le diagnostic des systèmes continus distribué. Dans ce cadre, le système se décompose en un ensemble de sous-systèmes en fonction de contraintes fonctionnelles, géographiques ou de confidentialité. La thèse se divise principalement en deux parties :

- la première partie cherche à mettre à lumière, à partir des modèles structurels obtenus au niveau des sous-systèmes, les redondances qui généreront des tests de diagnostic pertinents au niveau du système global,
- la deuxième partie vise à formuler et résoudre le problème d'optimisation lié au choix d'un sous-ensemble de tests de diagnostic au niveau des sous-systèmes permettant une diagnosticabilité maximale pour le système global.

La première partie utilise le concept d'ensemble minimal structurellement surdéterminé guidé par les fautes (Fault-Driven Minimal Structurally Overdetermined Set ou FMSO set). Ce concept est introduit dans la thèse. Il s'agit d'un sous-ensemble d'équations du modèle avec une redondance minimale à partir de laquelle

une RRA sensible à un ensemble de fautes peut être obtenu. Deux solutions pour générer des ensembles FMSO pour le système global sont présentées, d'une part dans un cadre décentralisé avec des superviseurs imbriqués suivant une hiérarchie ; d'autre part dans un cadre totalement distribué. Ces solutions sont basées sur les propriétés des ensembles FMSO au niveau des sous-systèmes qui sont présentées dans la thèse. La deuxième partie pose un problème d'optimisation dans le cadre d'une recherche heuristique et propose trois solutions basées sur un algorithme A* itératif combiné avec une fonction capable d'évaluer si un ensemble FMSO au niveau global peut être obtenu à partir des ensembles FMSO locaux sélectionnés. Les concepts introduits dans la thèse et les résultats sont appliqués à deux cas d'étude industriels. Le premier est une usine de désalinisation. Le second est un système de détermination et de contrôle d'attitude pour un satellite en orbite basse.

Abstract

The recent development of technological systems implies a high complexity of behaviors for today's systems. An answer to the increased system's complexity is to look at them as a multitude of heterogeneous subsystems and develop distributed techniques to control and manage them. This raises a number of problems. Firstly, as the size and number of components increase, so does the number of fault occurrences that may drive the system to undergo critical failures. Fault detection and isolation (FDI), maintenance and repair are an increasing part of the operational everyday's tasks and they impact drastically the total cost of final products.
This thesis focuses on fault detection and isolation. Among the different methods to generate diagnosis tests by taking advantage of analytical redundancy, this thesis adopts the so-called parity space approach based on analytical redundancy relations (ARRs). Given a model of the system in the form of a set of differential equations, ARRs are relations that are obtained from the model by eliminating non measured variables. This can be performed in an analytical framework using elimination theory but another way of doing this is to use structural analysis. Structural analysis is based on a structural abstraction of the model that only retains a representation of which variables are involved in which equations. Despite the rusticity of the abstract model, structural analysis provides a set of powerful tools, relying on graph theory, to analyze and infer information about the system. Interestingly, it applies indifferently to linear or nonlinear systems. The goal of this thesis is to develop effective techniques based on structural analysis for diagnosis of distributed continuous systems. In this framework, the system is decomposed into a set of subsystems according to functional, geographical or privacy constraints. The thesis is organized in two parts:

- highlighting the redundancies that are built into the global structural model and that can be used to generate diagnosis tests starting from the redundancies existing in the subsystem's models,

- formulating and solving the optimization problem linked to the choice of a subset of diagnosis tests at the subsystems level that can lead to a set of diagnosis tests achieving maximum diagnosability for the global system.

The first part takes benefit of the concept of Fault-Driven Minimal Structurally Overdetermined Set (FMSO set) that is introduced in the thesis. An FMSO set determines a subset of equations of the model with minimal redundancy from which an ARR sensitive to a set of faults can be obtained. Two solutions for generating FMSOs for the global system are presented, in a decentralized framework with supervisors at each level of a hierarchy and in a totally distributed framework. These are based on the properties of the FMSO sets for the subsystems in relation to those of the global system derived in the thesis.

The second part formulates the optimization problem in a heuristic search framework and proposes three solutions based on iterating an A* algorithm combined with a function able to assess whether a global FMSO set can be achieved from the selected local FMSO sets. The concepts introduced in the thesis and the results are applied to the case study of a Reverse Osmosis Desalination Plant and a Spacecraft Attitude Determination and Control System of a Low Earth-Orbit Satellite.

Acknowledgments

Firstly, and foremost, I would like to thank God for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily.

Secondly, I would like to express my sincere gratitude to my advisors Dr. Louise and Dr. Elodie for the continuous support of my PhD study and related research, for their patience, motivation, and for kindly receiving me in the beloved DISCO – LAAS team. In the same way, to my advisor Dr. Javier for encouraging my research and for allowing me to grow as a research scientist. I have great pleasure in acknowledging my gratitude to my colleagues and fellow research scholars, especially for my friend John. For this dissertation I would like to thank my reading committee members: Dr. Vincent Cocquempot, Dr. Addison Rios and Dr. Antonio Moran for their time, interest, and helpful comments.

Last but not the least, I would like to thank Joanna for all her love and support in these superb years together and my parents and brother for supporting me spiritually throughout this thesis and my life in general.

Gustavo
August 2017

Contents

List of Figures

List of Tables

List of Symbols

Z	The set of known (or measured) variables of a system
z	The vector of known (or measured) variables of a system
X	The set of unknown variables of a system
x	The vector of unknown (or measured) variables of a system
F	The set of faults of a system
f	The vector of faults of a system
Σ	The set of all equations of a system
$\mathcal{M}$	A complete matching
arr	An Analytical Redundancy Relation
$ARR_{j,i}$	A set of Analytical Redundancy Relationes for the subsystem $\Sigma_{j,i}$
Σ^+	The over-determined part of a system represented by Σ
Σ^0	The just-determined part of a system represented by Σ
Σ^-	The under-determined part of a system represented by Σ
ρ_Σ	The structural redundancy of a system Σ
$\mid \Sigma \mid$	The cardinality of a set Σ
X^+	The over-determined set of unknown variables of a system
X^0	The just-determined set of unknown variables of a system
X^-	The under-determined set of unknown variables of a system
MSO	A Minimal Structurally Overdetermined (MSO) set
TES	A Test Equation Support (TES)
$MTES$	A Minimal Test Equation Support (MTES)
φ	A Fault-Driven Minimal Structurally Overdetermined (FMSO) set
ψ	A Clear Minimal Structurally Overdetermined (CMSO) set
Φ	The set of Global FMSO sets of a system
X^l	The set of local unknown variables of a system
X^s	The set of shared unknown variables of a system
Φ^l	The set of local FMSO sets of a system
Φ^s	The set of shared FMSO sets of a system
Ψ^l	The set of local CMSO sets of a system
Ψ^s	The set of shared CMSO sets of a system
Φ^c	The set of compound FMSO sets of a system
Γ	The function that gives the set of successors of a node in a graph
$\Delta_{j,i}$	The set of all subsystems of level $j-1$ that have links to subsystem i at level j
$\Pi_{j,i}$	The set of additional equations for system $\Sigma_{j,i}$
$E_{j,i}$	The set of equations included in the shared FMSO and CMSO of system $\Sigma_{j,i}$
n	A node in a star-like algorithm

$f(n)$	The fscore of the node n
$g(n)$	The gscore of the node n
$h(n)$	The hscore of the node n
$\mathcal{I}_S$	The isolability degree of the set of FMSO (CMSO) sets to form the matrix S
$\mathcal{A}_i$	The set of ambiguity sets at step i

Part I

Preliminaries

Introduction

Global Overview and Motivation

One of the defining characteristics of modern societies is the ubiquitousness of large-scale technological systems which exhibit high complexification of behaviors. Systems become more and more pervasive and interconnected, so managing them in a global way becomes increasingly difficult and a solution is then to look at them as a multitude of heterogeneous subsystems and develop decentralized or distributed methods. At the same time, the requirements for reliability, availability and security are growing significantly: the prevention of serious engineering breakdowns is then becoming a crucial part of the development step of the systems and fault detection and isolation (FDI) are becoming a major issue. In addition, the development of such complex systems usually relies on a system engineering process that starts with system level requirements which are then functionally decomposed into subsystems.

These needs motivate the work of this thesis: develop efficient decentralized and distributed diagnosis methods that smoothly integrate with the system engineering process to detect and isolate faults impacting system continuous dynamics and that can cope with large interconnected systems.

Decentralized and distributed diagnoses imply separated local diagnosers (LD) for each subsystem. Each of these diagnosers works with local models of their respective subsystems, with no model sharing.

Two proposals are presented in this work. The first proposal is a decentralized fault diagnosis design method for systems that have some constraints as confidentiality, distance or limited access to some information and need an architecture organized in hierarchical supervisory levels. The second proposal focuses on distributed systems that do not require the inclusion of supervisory levels to coordinate the decision of the local diagnosers and for which the goal is to minimize communication links.

Main Objectives

The goal of this thesis is to develop effective techniques based on structural analysis for decentralized and distributed diagnosis of continuous systems. In this framework, the system is decomposed into a set of subsystems according to functional, geographical or privacy constraints.

The first objective of this thesis is to develop mathematical concepts that allow the design of efficient residual generators based on structural analysis techniques. Properties and computation procedures are formally defined. The notion of Fault-Driven Minimal Structurally Overdetermined (FMSO) set is introduced as the corner stone of the design of residual generators. Likewise, the concept of Clear

Minimal Structurally Overdetermined (CMSO) set is introduced as a complement of FMSO sets.

With the first objective fulfilled, the main goal is to decentralize and distribute the diagnosis approach according to system specifications, whereby the second objective of this thesis work is to formulate the design of the local diagnosers as an optimization problem: goal, criteria and constraints are formally defined. Then the goal is to propose relevant methods to solve the optimization problem. For the decentralized architecture, the goal is to select among the FMSO sets generated for each subsystem along the hierarchy in order to maximize the isolability degree. For the distributed architecture, the goal is to build a minimal set of global FMSO sets that guaranty maximal detectability and isolability, starting from shared FMSO/CMSO sets available for each subsystem and minimizing the number of tests and the interactions between subsystems. The common feature of the proposed algorithms is that they are cast in a heuristic search optimization framework.

Finally, the concepts and procedures introduced in the thesis and the results are applied to the case studies of a Spacecraft Attitude Determination and Control System of a Low Earth-Orbit Satellite and a Reverse Osmosis Desalination Plant.

Main Contributions

The main contributions of this thesis can be summarized in terms of contributions in structural analysis:

- The concept of Fault-Driven Minimal Structurally Overdetermined Set (FMSO set) is introduced. An FMSO set determines a subset of equations of the model with minimal redundancy from which an ARR sensitive to a set of faults can be obtained. The concept of Clear Minimal Structurally Overdetermined (CMSO) set, complementary of FMSO set, is also defined.

- Two solutions for generating FMSO sets for the global system are developed, in a decentralized framework with embedded supervisory levels and in a totally distributed framework. Important notions and related properties are presented. A proposition that states the conditions for which a union of shared FMSO/CMSO sets originating from different subsystems forms a global FMSO set also named compound FMSO set is presented. A compound FMSO set has a specific AND/OR tree structure.

- The third contribution is the formulation of two optimization problems, for the decentralized and for the distributed cases, in a heuristic search framework. The work proposes solutions based on A*-like algorithms combined with a function able to assess whether a global FMSO set can be achieved from the selected local FMSO sets.

Thesis Structure

This thesis is organized in five main parts. The first part presents preliminary information about the thesis context. The second part presents the operational procedures for the implementation of a decentralized and of a distributed fault diagnosis systems. The third part proposes to use A*-like algorithms for optimal test selection. The fourth part shows the application of the most important algorithms developed and the last part presents conclusions and perspectives. The content of each part is further explained below.

Part I is organized as follows.
In Chapter 1, a general framework for model based diagnosis methods is given. Then, the structural analysis and structural representations are presented for designing analytical redundancy relations. Generation of Minimal Structurally Overdetermined (MSO) sets and Minimal Test equation Support (MTES) are discussed and the concept of Fault-Driven Minimal Structurally Overdetermined (FMSO) sets is introduced as the corner stone of the design of residual generators in this thesis.

Chapter 2 introduces presents and compares centralized, decentralized and distributed diagnosis architecture as understood in this thesis. The diagnosis architecture describes which information is exchanged between the different components of the plant, the controller and the modules implementing the fault diagnosis function. Some important notions and properties are presented.

Part II is organized as follows.
Chapter 3 presents the operational procedure for the implementation of a decentralized fault diagnosis system considering the definitions given in Chapters 1 and 2. The procedure aims at finding for each level of a hierarchy the subset of local FMSO sets. This procedure can be used for what is called the *isolation on request* strategy.

Chapter 4 presents the operational procedure for the implementation of a distributed fault diagnosis system considering the definitions given in Chapters 1 and 2. First, the algorithm for the computation of all global FMSO sets based on local information only is presented and next an algorithm for distributed generation of an optimized set of global FMSO sets is introduced.

Part III includes Chapter 5 only and proposes to use A*-like algorithms to solve the different variants of the test selection problem. The goal is to optimally select the best FMSO sets for each subsystem to achieve the best possible fault detectability and isolability. In a distributed architecture, one also aims at minimizing communication between subsystems.

Part IV is organized as follows.
Chapter 6 presents the application of the decentralized fault diagnosis method proposed in Chapter 3 on the Attitude Determination and Control System of a real case study of Low Earth Orbit satellite. A decentralized decomposition of the LEO satellite into 2 subsystems and one supervisory level is proposed.

In Chapter 7, an application of the distributed fault diagnosis method pro-

posed in Chapter 4 is presented on a Reverse Osmosis Desalination Plant. Three subsystems are defined and the computation of global FMSO sets based on local information only is presented. The second part of this chapter consists in applying one of the algorithms presented in Chapter 5. The distributed generation of an optimized set of global FMSO sets is thus tested.

Part V ends this manuscript with Chapter 8 that proposes conclusions for this thesis work and recommendations for future work.

Publications

As a result of this thesis the following publications were produced:

- C.G. Pérez, L. Travé-Massuyès, E. Chanthery and J. Sotomayor (2015). *Decentralized diagnosis in a spacecraft attitude determination and control system.* Advanced Control and Diagnosis (ACD 2015), 12th European Workshop ACD 2015.

- C.G. Pérez, L. Travé-Massuyès, E. Chanthery and J. Sotomayor (2015). *Decentralized diagnosis in a spacecraft attitude determination and control system.* Journal of Physics: C. S., 2015 , 659 (2015) 012054.

- Carlos Gustavo Pérez, Elodie Chanthery, Travé-Massuyès and Javier Sotomayor (2016).*Fault Driven Minimal Structurally Overdetermined Set in a Distributed Context.* 27th International Workshop on Principles of Diagnosis: DX2016.

- R. Rivas-Pérez, J. Sotomayor-Moriano, C.G. Pérez-Zuñiga (2017). *Adaptive Expert Generalized Predictive Multivariable Control of Seawater RO Desalination Plant for a Mineral Processing Facility.* IFAC 2017 World Congress, Journal IFACPapersOnline.

- C.G. Pérez-Zuñiga, E. Chanthery, L. Travé-Massuyès, J. Sotomayor (2017). *Fault Driven Structural Diagnosis Approach in a Distributed Context.* IFAC 2017 World Congress, Journal IFACPapersOnline.

Chapter 1

A Framework for Model Based Diagnosis Methods

1.1 Introduction

Since about the middle of the previous century, the automation of the operation and the design of industrial processes has increased progressively. The expanding process automation was caused by an increasing demand on product quality, the independence of process operation from the presence of human operators, and the relief of operators from monotonous and heavy tasks, as well as by rising wages [Isermann 2011]. In recent decades, the complexity acquired by the industrial processes and their corresponding automation demand an adequate supervision and fault diagnosis of their technical processes in order to maintain safety standards in their operations and quality standard in their products, taking appropriate actions to avoid damage or accidents. The deviations from normal process behaviour result from faults, which can be have many causes. They may result sooner or later in malfunctions or failures if no counteractions are taken.

Fault diagnosis has been studied in two different fields, automatic control denoted by Fault Diagnosis and Isolation (FDI) and Artificial Intelligence (AI). These two distinct and parallel research communities have been using model-based diagnostic (MBD) techniques. Since the early 1970s, MBD techniques have developed remarkably to achieve satisfactory results in multiple applications in industrial processes and automatic control systems such as terrestrial, marine and aeronautical navigation systems, robots, transport systems, power systems, manufacturing processes or process control systems.

The diagnostic tasks can be summarized as follows [Blanke 2006]:

- **Fault detection:** Identifying deviations of the current system behaviour from the nominal behaviour, which is possible without a list of all possible faults.
- **Fault Isolation:** Localizing the faulty behavior to a certain (set of) components.
- **Fault Identification:** Classifying the fault as being of a particular type.

Without information about the faults and about the way in which the faults affect the system, no fault isolation and identification is possible. In order to identify the fault, fault models have to be known.

1.2 Principles of Model Based Diagnosis

In contrast to rule based methods, model based diagnosis (MBD) utilizes a model of the system together with observations from the real system to detect and isolate faults. Basing diagnosis decisions on a system model can address some of the crucial scalability and structural robustness issues limiting rule based diagnosers.

Actually, there are different MBD techniques applied for different purposes: these techniques have in common the explicit use of a *process model* and the information collected and processed on-line during the system operation. The major difference between the model-based fault diagnosis schemes lies in the form of the adopted *process model* that induces different algorithms [Ding 2008]. The occurrence of a fault is captured by discrepancies between the observed behavior and the behavior that is predicted by the *process model*.

In the last decades many investigations have been made using analytical approaches, based on quantitative models. The idea is to generate signals that reflect inconsistencies between nominal and faulty system operation. Such signals, termed residuals, are usually generated using analytical approaches, such as observers [Edwards 2000], [Chiang 2012], parameter estimation [Isermann 2006],[Isermann 2011] or parity equations [Gertler 1991], [Gertler 2000], [Butt 2012] based on analytical redundancy.

Computational complexity of multiple fault diagnosis is one of the well-known problems that needs to be tackled in order to deploy real-world applications of MBD [Chittaro 2004]. Equally, as complexity of the system increases, greater is the difficulty of designing efficient diagnosis systems. An alternative that is receiving greater acceptance from the academic and industrial world is the decentralization or distribution of the systems. Then, an important task is the design of detection and isolation tests in order to overcome faults quickly by obtaining fast and correct diagnostics. This can be understood as to obtain the complete set of over-constrained subsystems, as these subsystems carry the redundancy necessary to design fault tests. These fault tests can be deduced by means of *Structural Analysis* which is a tool that, despite its simplicity, provides important properties for the design of fault diagnosis systems and can be used for linear as well non linear systems [Armengol 2009], [Blanke 2006], [Bregon 2014].

Continuous systems are usually described by differential equations or transfer functions. With these models, the principle of consistency-based diagnosis can be illustrated by the scheme shown in Figure 1.1. The model is used to determine, for the measured input sequence $u = \langle u(t-k), ..., u(t) \rangle$, the model output sequence $\hat{y} = \langle \hat{y}(t-k), ..., \hat{y}(t) \rangle$. The consistency of the system with the model can be checked at every time t by determining the difference:

$$r(t) = y(t) - \hat{y}(t), \tag{1.1}$$

where $y(t)$ is the output of the real system.

$r(t)$ is called a *residual*. In the faultless case, the residual vanishes or is statisti-

cally zero. A non-vanishing residual indicates the existence of a fault. As shown in Figure 1.1 diagnostic algorithms for continuous systems generally consist of three steps:

- The model and the I/O pair are used to determine residuals, which describe the degree of consistency between the plant and the model behaviour.
- The residuals are evaluated.
- Finally, threshold generator is used to decide whether the residuals can be considered zero. An alarm may be triggered accordingly.

  In all steps, model uncertainties, disturbances and measurement noise have to be taken into account.

  Residual-based fault diagnosis systems make use of *analytical redundancy*: the model is an integral part of the diagnostic system and the residual is found by using more than one way for determining the output variables y. This procedure avoids physical redundancy where more than one sensor are used for measuring the same variable to get fault indicators.

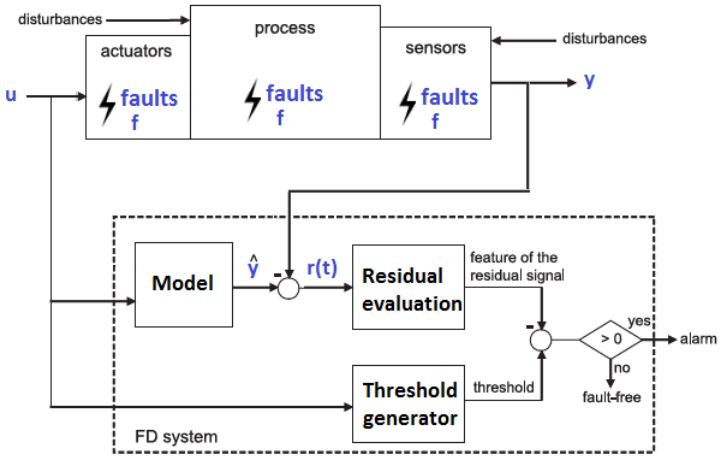


Figure 1.1: Residuals in fault detection systems.

1.3 Structural Analysis

1.3.1 Introduction

Structural analysis allows to obtain structural models that are very useful for the design of MBD systems. The main assumption is that each system component is described by one or several constraints, thereby, violation of at least one constraint indicates that the system component is faulty. Structural properties lead to analytical redundancy based residuals [Blanke 2006]. The structural model of a system is

an abstraction of its behaviour model in the sense that only the structure if the constraints, i.e. the existence of links between variables and parameters is considered and not the constraints themselves.

Fault tests obtained by structural analysis can be turned into parity relations or analytical redundancy relations (ARR) and are designed off-line. The fault detection system then checks on-line the consistency of the observations with respect to every of these tests.

Structural analysis constitutes a general framework to provide information when the system becomes complex, e.g., dividing complex non linear systems into smaller solvable problems [Skovmose 2006]. The main aim of the structural approach is to identify the subsets of equations of the model which include redundancy. Structural analysis is applicable to a large class of models without details of parameter values and to reduce computational complexity of identifying model redundancy in differential algebraic models by means of efficient graph-based tools [Cassar 1997]. Originally developed for the decomposition of large systems of equations for their hierarchical resolution, structural analysis was successfully adopted for some decades by the Fault Detection and Isolation (FDI) community [Cassar 1997, Patton 2000, Travé-Massuyès 2006, Blanke 2006, Krysander 2010].

Since only structural information is used, this approach applies to large scale systems described by a great number of variables, even when their analytical models are not precisely known [Düstegör 2006, Cassar 1997]. The structural decomposition of a system can constitute a good way to analyze the redundancy of the knowledge available about the system. This redundancy is used to detect and isolate faults.

Structural properties determine monitoring abilities. The concept of complete matching on a graph is a key concept.

Let the system description consist of a set of n_e equations involving a set of variables partitioned into a set Z of n_Z known (or measured) variables and a set X of n_X unknown (or unmeasured) variables. We refer to the vector of known variables as z and the vector of unknown variables as x. The system may be impacted by the presence of n_f faults that appear as parameters in the equations. The set of faults is denoted by F and we refer to the vector of faults as $\boldsymbol{f}$.

Definition 1 (System)**.** *A system, denoted $\Sigma(z, x, \boldsymbol{f})$ or Σ for short, is any set of equations relating z, x and $\boldsymbol{f}$. The equations $e_i(z, x) \subseteq \Sigma(z, x, \boldsymbol{f})$, $i = 1, \ldots, n_e$, are assumed to be differential or algebraic in z and x.*

Example 1.3.1. *Consider the illustrative system shown in Table 1.1 for which the model $\Sigma(z, x, \boldsymbol{f})$ is composed of six equations e_1 to e_6 relating:*

- *the known variables $Z = \{z_1, z_2\}$,*
- *the unknown variables $X = \{x_1, x_2, x_3, x_4, x_5\}$ and*
- *the set of system faults $F = \{f_1, f_2, f_3\}$*
- *besides au,a,b are constant parameters.*

Relation	Expression
e_1	$\dot{x}_3 = e^{x_3} - au$
e_2	$x_3^2 = b\dot{x}_4 + f_1$
e_3	$z_1 = x_4$
e_4	$z_2 = x_1 + a^2 + x_4$
e_5	$\dot{x}_1 = e^{x_2} + x_5$
e_6	$\dot{x}_3 = x_4 + b + f_2$

Table 1.1: Differential-algebraic illustrative system.

1.3.2 Structural Representations

The structural representation of Example 1.3.1 is the biadjacency matrix given in Table 1.2.

Equation	Unknown					Known		Faults	
	x_1	x_2	x_3	x_4	x_5	z_1	z_2	f_1	f_2
e_1			X						
e_2			X	X				X	
e_3				X		X			
e_4	X			X			X		
e_5	X	X			X				
e_6			X	X					X

Table 1.2: Structural representation of the illustrative example.

In this representation all unknowns variables: $x_1, ..., x_5$ are considered to be signals. There is an "X" in position (i, j) in the biadjacency matrix if x_j or any of its time-derivatives appears in equation e_i. More generally an "X" indicates that a variable appears in equation e_i.

The structure model is conveniently represented as a graph. Here a bipartite graph is used in order to represent the information about which variables are involved in each equation and numerical values and analytical expressions are thereby ignored. The bipartite graph associated with the example 1.3.1 is shown in Figure 1.2.

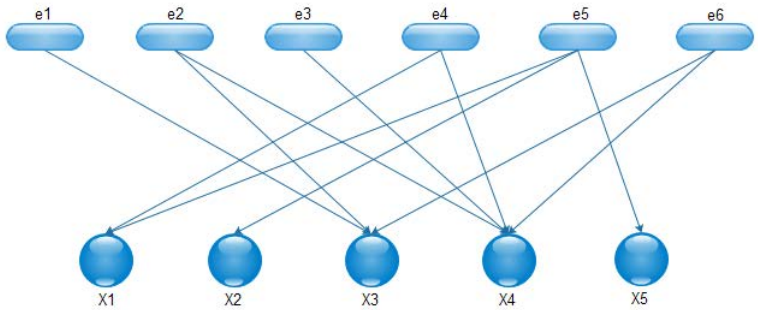


Figure 1.2: The bipartite graph for the illustrative example 1.3.1.

Specifically, a bipartite graph can be used to represent which unobserved variables are involved in the equations modelling the system and infer a possible path for variable substitution. The task of finding redundancy in a model can thus be reformulated as a graph-theoretical problem that can be solved with efficient methods developed for bipartite graphs in the graph theory [Dulmage 1958]. These methods are free from numerical problems and have in general lower computational complexity than algebraic elimination methods [Khorasgani 2015].

The *structural model* of the system $\Sigma(z, x, f)$, also denoted with some abuse by $\Sigma(z, x, f)$ or Σ in the following, can be obtained abstracting the functional relations. This abstraction leads to a bipartite graph $G(\Sigma \cup X \cup Z, A)$, or equivalently to $G(\Sigma \cup X, \mathcal{A})$, where $\mathcal{A} \subseteq A$ and $\mathcal{A}$ is a set of edges such that $a(i, j) \in \mathcal{A}$ iff variable x_i is involved in equation e_j.

A bipartite graph with vertices partitioned into the sets Σ and X and edges $\mathcal{A}$ is written $G = (\Sigma \cup X, \mathcal{A})$. Let the two vertex sets be explicitly ordered, let us say $\Sigma = \{e_1, e_2, ..., e_m\}$ and $X = \{x_1, x_2, ..., x_n\}$. Then a *biadjacency matrix* for a bipartite graph G is the $m \times n$ matrix $\mathcal{A}$ defined by:

$$\mathcal{A}_{i,j} = \begin{cases} X & \text{if } e_i \text{ and } x_j \text{ are adjacent} \\ 0 & \textit{otherwise} \end{cases} \tag{1.2}$$

The biadjacency matrix for the example 1.3.1 is:

$$\mathcal{A} = \left\{ \begin{array}{c|ccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline e_1 & & & X & & \\ e_2 & & & X & X & \\ e_3 & & & & X & \\ e_4 & X & & & X & \\ e_5 & X & X & & & X \\ e_6 & & & X & X & \end{array} \right. \tag{1.3}$$

It can be shown [Blanke 2006] that ARRs can be derived from so-called complete matchings between X and Σ on the bipartite graph $G(\Sigma \cup X, \mathcal{A})$.

Definition 2 (Matching)**.** *: A matching between X and Σ is a subset of $\mathcal{A}$ such that no vertex in $X \cup \Sigma$ is incident with more that one edge of the matching.*

Definition 3 (Complete Matching)**.** *: A complete matching between X and Σ is a matching covering every vertex in X.*

A complete matching between X and Σ is denoted by $\mathcal{M}(X, \Sigma)$, or $\mathcal{M}$ when there is no ambiguity. $\mathcal{M}(X, \Sigma)$ provides a way to identify the paths to calculate the unknown variables from the measured or shared variables.

Figure 1.3 illustrates an example of complete matching indicated by bold red edges in the corresponding bipartite graph. Here, for instance, the unknown variables x_2 and x_3 are matched to the equations e_2 and e_5 respectively. Besides, equations e_3 and e_4 are redundant because these are not involved in any complete

matching. This means that e_3 and e_4 are not needed to calculate the unknown variables and that can be used to check for consistency.

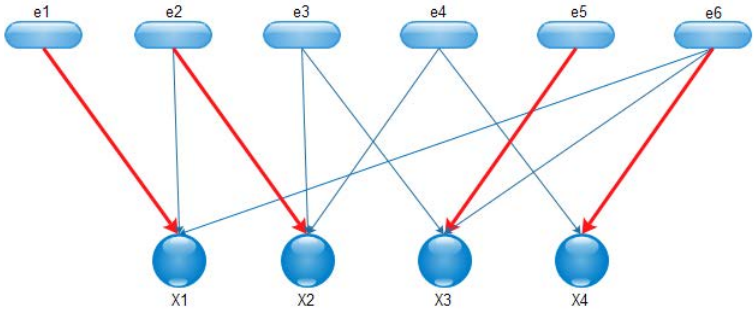


Figure 1.3: Example of complete matching.

1.4 Analytical Redundancy Relations

1.4.1 Definition

The main concept for residual generation in the case of continuous variable systems is analytical redundancy. Analytical redundancy relations (ARR) are equations that are deduced from an analytical model and only involve measured variables. Analytical redundancy relations are consistent in the absence of faults, and can thus be used for residual generation.

ARRs are static or dynamic constraints that capture the temporal behavior of known variables with the system operating in nominal conditions. Once ARRs are designed, the fault detection procedure checks at each time whether they are satisfied or not, and when not, the fault isolation procedure identifies the system components which are to be suspected.

Definition 4 (ARR for $\Sigma(z, x, f)$)**.** *Let $\Sigma(z, x, f)$ be a system. Then, a relation $arr(z, \dot{z}, \ddot{z}, ...) = 0$ is an Analytical Redundancy Relation (ARR) for $\Sigma(z, x, f)$ if for each z consistent with $\Sigma(z, x, f)$ the relation is fulfilled.*

Definition 5 (Residual Generator for $\Sigma(z, x, f)$)**.** *A system taking a subset of the variables z as input, and generating a scalar signal arr as output, is a residual generator for the model $\Sigma(z, x, f)$ if, for all z consistent with $\Sigma(z, x, f)$, it holds that $\lim_{t\to\infty} arr(t) = 0$.*

ARRs can be used to check if the measured variables z are consistent with the system model and as the basis of residual generators used for diagnosis purposes.

Several methods for computing sets with structural redundancy in $G(\Sigma \cup X, \mathcal{A})$ are based on the Dulmage-Mendelsohn DM canonical decomposition (Murota, 2000; Dulmage and Mendelsohn, 1958). the Dulmage–Mendelsohn decomposition is a partition of the vertices of a bipartite graph into subsets, with the property that two adjacent vertices belong to the same subset if and only if they are paired with

each other in a perfect matching of the graph. A perfect matching is a matching which matches all vertices of the graph. That is, every vertex of the graph is incident to exactly one edge of the matching.

The biadjacency matrix in Figure 1.4 shows Dulmage-Mendelsohn (DM) canonical decomposition of a bipartite graph $G(\Sigma \cup X, \mathcal{A})$. The light blue-shaded areas contain ones and zeros, while the white areas only contain zeros. The diagonal line represents a maximal matching in the graph $G(\Sigma \cup X, \mathcal{A})$ where the rows and columns are rearranged. The model Σ is partitioned in three parts: Σ^-, Σ^0 and Σ^+ and the unknowns are partitioned accordingly. The set Σ^+ is exactly the set of equations $e \in \Sigma$ such that for any maximum size matching there exists an alternating path between at least one free equation vertex and e. The set X^+ is the set of vertices adjacent to at least one vertex in Σ^+. The set X^- is exactly the set of variable vertices $x \in X$ such that for any maximum size matching there exists an alternating path between at least one free variable vertex and x. The set Σ^- is the set of vertices adjacent to at least one vertex in Σ^-. The remaining sets of vertices in Σ and X are Σ^0 and X^0 respectively [Blanke 2006]. In summary, the DM canonical decomposition results in a partition of the system model Σ into three parts:

- The **structurally over-determined** (SO) part Σ^+ with more equations than unknown variables,
- The **structurally just-determined** part Σ^0, and
- The **structurally under-determined** part Σ^- with more unknown variables than equations.

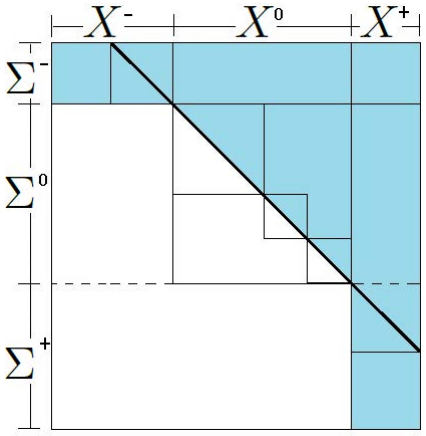


Figure 1.4: The Dulmage-Mendelsohn decomposition of a model Σ.

Example 1.4.1. *Consider the model Σ of the example 1.3.1, through the Dulmage-Mendelsohn canonical decomposition of its respective incidence matrix in table 1.6.2, the following matrix is obtained:*

Equation	Unknown					Known		Faults	
	x_5	x_2	x_1	x_3	x_4	y	z	f_1	f_2
e_5	X	**X**	X						
e_4			**X**		X		X		
e_1				**X**					
e_2				X	**X**			X	
e_3					X	X			
e_6				X	X				X

Table 1.3: DM decomposition of the illustrative example.

For this example, with the notations of Figure 1.4, the sets are $\Sigma^+ = \{e_1, e_2, e_3, e_6\}$, $\Sigma^0 = \{e_4\}$ and $\Sigma^- = \{e_5\}$ and $X^+ = \{x_3, x_4\}$, $X^0 = \{x_1\}$ and $X^- = \{x_5, x_2\}$. Likewise, the bolded $\mathbf{X}$ denote a maximal matching in $G(\Sigma \cup X, \mathcal{A})$.

1.4.2 Analytical Redundancy via Structural Analysis

From the point of view of structural analysis, redundancy relations are subgraphs of the structure graph of the system, which are associated with complete matchings of the unknown variables associated with the over-constrained subsystem of the reduced bipartite graph. Redundancy relations are composed of alternated chains, which start with known variables and which end with non-matched constraints.

1.5 Residual Generation

1.6 Introduction

Using structural analysis, it is possible determine the part(s) of the system on which some ARRs can be generated to be calculated [Cocquempot 1998]. Obtaining ARRs for a system $\Sigma(z, x, \mathtt{f})$ involves the elimination of unknown variables, which can be inferred from structural analysis [Travé-Massuyès 2006]. ARRs are indeed known as the causal interpretation of minimal structurally overdetermined (MSO) sets [Krysander 2010]. One should notice that results obtained in a structural framework are a best case scenario: causality considerations, algebraic and differential loops, etc. ultimately define which structural redundancies can be used for the design of actual residual generators [Armengol 2009].

Definition 6 (Structural redundancy)**.** *The structural redundancy $\rho_{\Sigma'}$ of a set of equations $\Sigma' \subseteq \Sigma$ is defined as the difference between the number of equations in Σ' and the number of unknown variables involved in the equations.*

$$\rho_{\Sigma'} = \mid \Sigma' \mid - \mid X_{\Sigma'} \mid \tag{1.4}$$

where $X_{\Sigma'}$ is the set of unknown variables involved in Σ' and $\mid \Sigma' \mid$ is the cardinality of Σ' .

Definition 7 (Structurally overdetermined (SO))**.** *A set $\Sigma' \subseteq \Sigma$ of equations is SO if Σ' has more equations that unknowns variables.*

The structural redundancy of an SO set is positive. Let us notice that the structural redundancy of an arbitrary set of equations $\Sigma' \subseteq \Sigma$ may be positive, zero, or negative.

Proposition 1.6.1. *Consider two sets of equations $\Sigma' \subseteq \Sigma$ and $\Sigma'' \subseteq \Sigma$, then*

$$\rho_{\Sigma' \cup \Sigma''} = \rho_{\Sigma'} + \rho_{\Sigma''} + |X_{\Sigma'} \cap X_{\Sigma''}|. \tag{1.5}$$

Proof. By using 1.4: $\rho_{\Sigma'} =| \Sigma | - | X_{\Sigma'} |$ and $\rho_{\Sigma''} =| \Sigma | - | X_{\Sigma''} |$. Then:

$$\begin{aligned} \rho_{\Sigma' \cup \Sigma''} &= | \Sigma' \cup \Sigma'' | - | X_{\Sigma' \cup \Sigma''} | && (1.6)\\ &= | \Sigma' | \cup | \Sigma'' | -(| X_{\Sigma'} | + | X_{\Sigma''} | - | X_{\Sigma' \cap \Sigma''} |) && (1.7)\\ &= \rho_{\Sigma'} + \rho_{\Sigma''} + |X_{\Sigma'} \cap X_{\Sigma''}| && (1.8) \end{aligned}$$

□

The structurally overdetermined (SO) set is used now to define the ideas of proper structurally overdetermined (PSO) set, minimal structurally overdetermined (MSO) set and Minimal Test equations Support (MTES) [Krysander 2010]. Then the notion of Fault-Driven Minimal Structurally Overdetermined (FMSO) set is introduced. Following, these ideas are compared to generate analytical redundancy relations (ARR).

1.6.1 Generate Minimal Structurally Overdetermined (MSO) Sets

Definition 8 (Proper structurally overdetermined (PSO) set)**.** *A set of equations Σ is proper structurally overdetermined (PSO) set if $\Sigma = \Sigma^+$.*

Definition 9 (Minimal Structurally Overdetermined (MSO) set)**.** *A MSO set is a structurally overdetermined set whose no proper subset of Σ is structurally overdetermined [Krysander 2008a].*

A detailed comparison of some algorithms to generate MSO sets was done in [Armengol 2009].

- The MSO-Algorithm [Krysander 2008a] calculates MSOs by eliminating equations from the original set Σ^+ until the structural redundancy is 1. This efficient algorithm compute all the MSO sets given a structural model. Is based on a top-down search, beginning with the complete structurally overdetermined part Σ^+ of the system model, and recursively constructing the search tree by removing equations until all the MSO sets are found.[1]

[1]The latest version of this algorithm can be obtained as part of the Fault Diagnosis Toolbox of the authors E. Frisk and M. Krysander in https://faultdiagnosistoolbox.github.io/

- The CBMSOs-Algorithm [Gelso 2008] computes MSO sets by complete matching between equations and unknown variables. the algorithm was improved such that all possible MSO sets are found. This was accomplished by iteratively combining the set of MSO sets computed in the former algorithm.

- A modification of [Krysander 2008a] method was presented in [Rosich 2009a]. The objective of this modification is to guarantee that all the unknown variables in an MSO set can be easily computed when both, linear and non-linear equations, are involved in the MSO set.

It is necessary to note that there are differences between an ARR and an MSO set, the main difference is that the computation of the unknown variables must be explicitly ensured in an ARR whereas in an MSO set is not [Armengol 2009]. It can be concluded that an ARR is a particular case of an MSO set. Some other algorithms for generating ARRs are exposed below:

- In [Blanke 2006] one of the early algorithms to generate ARRs was proposed. This algorithm starts with a maximum matching in the set of unknown variables and subsequently a family of ARR is found by adding extra non-matched equations to the set of equations involved in the matching but the result is not complete because not all the possible ARRs are guaranteed.

- The SARR-Algorithm [Travé-Massuyès 2006] proceeds to successive elimination of unknown variables on the structural model equations to obtain the set of minimal SARRs. Contrary to MSO sets. SARRs account for causality indicating how to chain the equations to generate an actual RRA.

A conclusion drawn from these algorithms is that searching for all the MSO sets has exponential time complexity. For example, given a model Σ' with n equations and with structural redundancy $\rho_{\Sigma'}$, there are at most m PSO subsets, where m is given by the equation 1.9.

$$m = \sum_{k=n-\rho_{\Sigma'}+1}^{n} \binom{n}{k} \tag{1.9}$$

For a fixed order of structural redundancy ρ, the complete version of Algorithm in [Krysander 2008a] has order of $n^{\rho+1.5}$ time complexity.

Example 1.6.1. *Consider the benchmark problem of the two coupled tanks depicted in Figure 1.5 that provide a continuous water flow Q_0 to consumers [Armengol 2009].*

The components are tanks T_1 and T_2, controllers *PI* and *ON/OFF*, pump P_1, proportional valves V_b and V_o, level sensors my_1 and my_2, flow sensor mU_p.

The model $\Sigma(z, x, \mathtt{f})$ is composed of eleven equations e_1 to e_{11} relating the known variables $Z = \{my_1, my_2, mQ_p, mU_p, h_{1c}, mU_b, mU_0\}$, the unknown variables

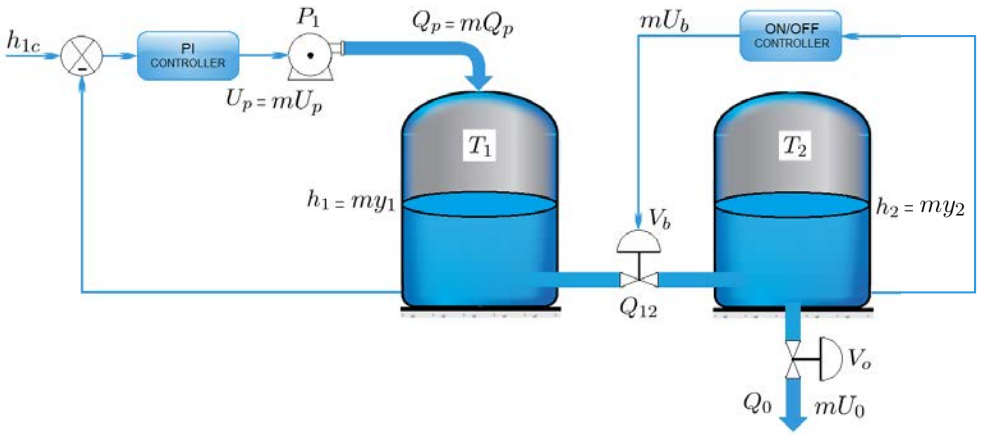


Figure 1.5: Two coupled tanks benchmark.

$X = \{h_1, h_2, Q_p, Q_0, Q_{12}, U_p\}$ and the set of system faults $F = \{f_1, f_2\}$ as given in Table 3.1.

A_1 and A_2 are the areas of the cylindrical tanks; K_p and K_i are parameters of the PI controller; C_{vb} and C_{vo} are the global hydraulic flow coefficients of the valves V_b and V_o; and mU_0 is the V_o valve position $\in \{1, 0\}$ provided by the user.

Table 1.4: Equations for the two tank system.

e_1 : $\dot{h}_1 = \dfrac{1}{A_1 + f_1}(Q_p - Q_{12})$

e_2 : $\dot{h}_2 = \dfrac{1}{A_2 + f_2}(Q_{12} - Q_0)$

e_3 : $Q_p = \begin{cases} U_p & \text{if } 0 < U_p < Q_{pmax} \\ 0 & \text{if } U_p \leq 0 \\ U_{pmax} & \text{if } U_p \geq Q_{pmax} \end{cases}$

e_4 : $Q_{12} = C_{vb} \cdot sgn(h_1 - h_2) \cdot \sqrt{\mid h_1 - h_2 \mid}$

e_5 : $Q_0 = C_{vo} \cdot \sqrt{h_2} \cdot mU_0$

e_6 : $U_p = K_p(h_{1c} - h_1(t)) + K_i \int (h_{1c} - h_1(t))dt$

e_7 : $mU_b = \begin{cases} 0 & \text{if } 0.09m < h_2 < 0.11m \\ 1 & \text{if } 0m < h_2 < 0.09m \end{cases}$

e_8 : $my_1 = h_1$

e_9 : $my_2 = h_2$

e_{10} : $mQ_p = Q_p$

e_{11} : $mU_p = U_p$

The structural representation of example 2.4.2 is the biadjacency matrix given in Table 1.6.2

The MSO set - Algorithm proposed finds 165 PSO sets and 46 MSO sets. The calculation of 46 MSO sets represents a high computational complexity considering that it is a small example that only includes two faults. This is because the concept of the MSO sets does not consider any discrimination to calculate the MSO sets

Equation	Unknown						Faults	
	h_1	h_2	Q_p	Q_0	Q_{12}	U_p	f_1	f_2
e_1	X		X		X		X	
e_2		X		X	X			X
e_3			X			X		
e_4	X	X			X			
e_5		X		X				
e_6	X					X		
e_7		X						
e_8	X							
e_9		X						
e_{10}			X					
e_{11}						X		

Table 1.5: Structural representation of two coupled tanks benchmark.

based on some set of fault support. A problem with this approach is that the number of MSO sets grows exponentially in the degree of redundancy of the model.

In large complex systems it might not be necessary (for diagnosis purposes) to construct and use all possible sets since there might exists a significantly smaller number of them with sufficient capability of distinguishing between different faults.

1.6.2 Minimal Test Equation Support (MTES)

As discussed above, the number of MSO sets grows exponentially in the degree of redundancy of the model. By including faults of interest, the resulting number of testable subsets as well as the computational complexity of finding them can be reduced drastically. Alternatively to the search for all MSO sets, [Krysander 2010] proposed to search for a smaller set of testable models called Test Equation Supports (TESs) which is a set of equations expressing redundancy specific to a set of considered faults. This set of faults is known as the test support (TS). An MTES and a minimal TS (MTS) are such that no proper subset is a TES and TS, respectively.

Definition 10 (Test support (TS))**.** *Given a set of equations Σ and a set of faults F_Σ, a subset of faults $\zeta \subseteq F_\Sigma$ is a test support if there exists a PSO set $\Sigma' \subseteq \Sigma$ such that $F_{\Sigma'} = \zeta$.*

Definition 11 (Minimal test support (MTS))**.** *A test support is a minimal test support (MTS) if no proper subset is a test support.*

Definition 12 (Test Equation Support (TES))**.** *A set of equations Σ is a Test Equation Support (TES) if $F_\Sigma \neq \emptyset$, Σ is PSO set and for any $\Sigma' \not\supseteq \Sigma$ where Σ' is a PSO set it holds that $F_{\Sigma'} \not\supseteq F_\Sigma$.*

Definition 13 (Minimal test equation support (MTES))**.** *A TES Σ is a minimal TES (MTES) if there exists no subset of Σ that is a TES.*

Example 1.6.2. *Consider the biadjancency matrix of the example 2.4.2. Using the algorithm proposed in [Krysander 2010], we can calculate the TES and MTES.*

TES	F_{TES}	ρ_{TES}
$\text{TES}_1 = \{e_1, e_2, e_3, e_4, e_5, ..., e_{11}\}$	$F_{TES_1} = \{1, 2\}$	$\rho_{TES_1} = \{5\}$
$\text{TES}_2 = \{e_1, e_3, e_4, e_5, ..., e_{11}\}$	$F_{TES_2} = \{1\}$	$\rho_{TES_2} = \{4\}$
$\text{TES}_3 = \{e_2, e_3, e_4, e_5, ..., e_{11}\}$	$F_{TES_3} = \{2\}$	$\rho_{TES3} = \{4\}$

Table 1.6: TES of the example 2.4.2.

MTES	F_{MTES}	ρ_{MTES}
$\text{MTES}_1 = \{e_1, e_3, e_4, e_5, ..., e_{11}\}$	$F_{MTES_1} = \{1\}$	$\rho_{MTES_1} = \{4\}$
$\text{MTES}_2 = \{e_2, e_3, e_4, e_5, ..., e_{11}\}$	$F_{MTES_2} = \{2\}$	$\rho_{MTES_2} = \{4\}$

Table 1.7: MTES of the example 2.4.2.

For instance, $\text{MTES}_1 = \{e_1, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$ focused in fault f_1 with redundancy = 4.

Whereas an MSO set is just overdeterminated and hence has redundancy 1, an MTES may have higher redundancy, as in the previous example where both MTES have redundancy 4. This may be an advantage to develop more powerful tests; however, for the decentralization /distribution problem studied in this thesis, the aim is to minimize the information shared by subsystems, hence the concept of Fault-Driven Minimal Structurally Overdetermined set is introduced.

1.6.3 Fault-Driven Minimal Structurally Overdetermined (FMSO) Sets

A Fault-Driven Minimal Structurally Overdetermined (FMSO) set φ is an MSO set of $\Sigma(z, x, \mathtt{f})$ whose fault support is not empty. In particular, an MTES of structural redundancy 1 is an FMSO set.

Let us define $Z_\varphi \subseteq Z$, $X_\varphi \subseteq X$, and $F_\varphi \subseteq F$ as the set of known variables, unknown variables involved in an FMSO set φ, and the set of faults in its fault support, respectively. We then have the following formal definition.

Definition 14 (FMSO set)**.** *A subset of equations $\varphi \subseteq \Sigma(z, x, \mathtt{f})$ is an FMSO set of $\Sigma(z, x, \mathtt{f})$ if (1) $F_\varphi \neq \emptyset$ and $\rho_\varphi = 1$ that means $|\varphi| = |X_\varphi| + 1$, (2) no proper subset of φ is overdeterminated.*

Now, let us define the concepts of *detectable fault*, and *isolable fault* applying the concept of FMSO sets.

Definition 15 (Detectable fault)**.** *A fault $f \in F$ is detectable in the system $\Sigma(z, x, \mathtt{f})$ if there is an FMSO set $\varphi \in \Phi$ such that $f \in F_\varphi$, where the set of fault of the system is denoted by F.*

The concept of isolation is based on determining the set of faults that can be isolated from a given fault.

Definition 16 (Isolable fault)**.** *Given two detectable faults f_j and f_k of F_i, $j \neq k$, f_j is isolable from f_k if there exists an FMSO set $\varphi \in \Phi$ such that $f_j \in F_\varphi$ and $f_k \notin F_\varphi$.*

We also define the concept of *Clear Minimal Structurally Overdetermined* (CMSO) set as an MSO set of $\Sigma(z, x, \boldsymbol{f})$) whose fault support is empty.

Definition 17 (CMSO set)**.** *A subset of equations $\psi \subseteq \Sigma(z, x, \boldsymbol{f})$ is a CMSO set of $\Sigma(z, x, \boldsymbol{f})$ if (1) $F_\psi = \emptyset$ and $\rho_\psi = 1$ that means $|\psi| = |X_\psi| + 1$, (2) no proper subset of ψ is overdeterminated.*

Example 1.6.3. *To illustrate these concepts, consider the example 2.4.2 to calculate the FMSO sets.*

For this example were found 42 FMSO sets, four less than MSO sets. Within them are two FMSO sets that have minimal redundancy and are focused on each of the faults of interest as shown in Table 1.8: $\varphi_1 = \{e_1, e_4, e_7, e_8, e_{10}\}$ focused on fault f_1 and $\{e_2, e_4, e_5, e_7, e_8\}$ focused on fault f_2 both with redundancy = 1.

FMSO set φ	F_φ	ρ_φ
$\varphi_1 = \{e_1, e_4, e_7, e_8, e_{10}\}$	$F_{\varphi_1} = \{1\}$	$\rho_{\varphi_1} = \{1\}$
$\varphi_2 = \{e_2, e_4, e_5, e_7, e_8\}$	$F_{\varphi_2} = \{2\}$	$\rho_{\varphi_2} = \{1\}$

Table 1.8: FMSO sets of the example 2.4.2 for f_1 and f_2.

Comparing the two FMSO sets found (both with redundancy = 1)with the two MTES obtained in the previous section (both with redundancy = 4), it is shown that using FMSO sets is more efficient for systems that may be too large or complex where it becomes imperative to develop decentralized or distributed diagnosis approaches that require minimizing data transfer.

1.7 Conclusion

Structural analysis enables one to investigate model redundancy by means of efficient graph-based tools and constitutes an interesting framework to provide information when the system becomes complex, regardless of the linear or non-linear nature of the system. Fault-Driven Minimal Structurally-Overdetermined (FMSO) set concept has been introduced. It can be directly used to construct one ARR or residual generator, as compared to MTES that lead to several. Therefore, it is possible to consider that FMSO sets represent a more practical solution in decentralized or distributed contexts in which communication must be minimized.

In this thesis work, two approaches will be presented for the design of the fault diagnosis system for large complex systems with constraints such as communication bandwidth or large geographic distribution: decentralized and distributed approaches.

Accordingly, the main objective of this thesis is the design first of a decentralized diagnosis system and next of a distributed diagnosis for continuous systems taking into account pre-existing constraints that may be functional and predefined subsystems optimizing the transmission of information.

To achieve this objective, the structural approach is used for generating Analytical Redundancy Relations, then the problem is the appropriate selection of Analytical Redundancy Relations focused on a set of interesting faults. Finally the work focuses on the problem of optimizing the choice of Analytical Redundancy Relations for each subsystem.

Chapter 2

Notions for Decentralized and Distributed Fault Diagnosis

Contents

2.1 Introduction

The architecture of fault diagnosis systems describes which information is exchanged between the different components of the plant, the controller and the modules implementing the fault diagnosis function.

Generally, design theories of fault-diagnosis systems focus on global system approaches [Patton 2000], [Isermann 2011], [Korbicz 2012] where the entire fault diagnosis system is loaded on a single computer, which is directly connected to the system to be monitored. All measurement information is available on this board and therefore, all algorithms have all the information available. However, there are important practical circumstances in which centralized diagnosis architectures can not be applied, for example existence of pre-existing constraints that may be functional, geographical or privacy-based (e.g. confidenciality, restricted availability of information by area). These can be found in aircraft and other transportation

systems, manufacturing processes, supply chain and distribution networks, power generation and other similar processes and systems. Decentralized or distributed diagnosis approaches must then be applied, where fault diagnosis algorithms and measurement information are distributed between different components.

Let notice also that centralized diagnosis solutions have several inherent shortcomings. First, if the centralized diagnoser fails, the system has to operate without diagnosis system (this is usually known as a single point of failure), and second, centralized solutions do not scale well as the size of the system increases [Gertler 1998]. These shortcomings justify the development of techniques of decentralized or distributed diagnosis.

Next section aims at describing each type of architecture in order to develop the two main non-centralized types of architectures in the following chapters. Related work on decentralized and distributed fault diagnosis is then discussed. The final section gives some important notions and properties.

2.2 Architectural Options for Fault Diagnosis Systems

Three categories of fault diagnosis architectures are usually considered according to the literature:

- Centralized architecture,
- Decentralized architecture,
- Distributed architecture.

These are explained and motivated below.

2.2.1 Centralized Diagnosis Architecture

A centralized diagnosis architecture gathers data into a centralized fault diagnosis system which computes so called global diagnosis. Figure 2.1 shows an example of this kind of architecture.

According to the complexity of the system, this solution can be simple and easy to implement. There is no subdivision of the diagnostic system and thus, theoretically, there are no communication problems. However it requires to explicitly build a global model of the system, which is usually not possible for large systems for many reasons, e.g., when the system covers a large geographic area and the measurements are distributed so that they cannot be directly wired to the processing computer. Moreover, there are contexts where a centralized architecture, even if feasible, would be undesirable because of several factors including size, robustness and security issues, e.g., aircraft and other transportation systems, large-scale energy or industrial plants, power generation, etc. Distributing and decentralizing diagnosis are two solutions to cope with these difficulties.

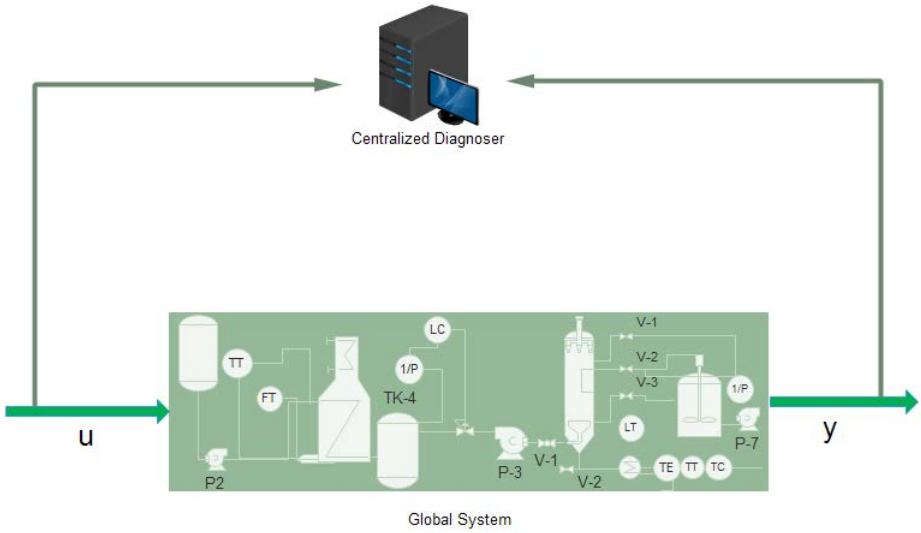


Figure 2.1: Centralized diagnosis architecture.

2.2.2 Decentralized Diagnosis Architecture

A decentralized diagnosis architecture assumes a decomposition of the process in subsystems each with its corresponding local diagnoser, the diagnostic task is coordinated by a supervisory diagnoser to ensure the consistency between local diagnosers. The supervisory levels may be more than one, depending on how the subsystems are recursively grouped together, hence forming a supervisory hierarchy of embedded subsystems. An illustration of such hierarchy is given in Figure 2.2. The diagnosis system is decomposed into different local diagnosers which refer to the subsystems of the complete system. The local diagnosers process local measurements independently one from the others.

2.2.3 Distributed Diagnosis Architecture

A distributed diagnosis architecture assumes a decomposition of the process into subsystems each with its corresponding local diagnoser, with similar functions and with possible communication between them. This communication must be properly designed so that the local diagnoses are globally consistent. Here are two possible design approaches:

1. The diagnostic system is designed as a unique entity and the resulting algorithm is distributed over different components to cope with the computational effort needed.

2. Considering the constraints of the system, local diagnostic systems are designed independently, considering the communications between them as shown in Figure 2.3, until reaching the same diagnosis as with a centralized diagnosis design.

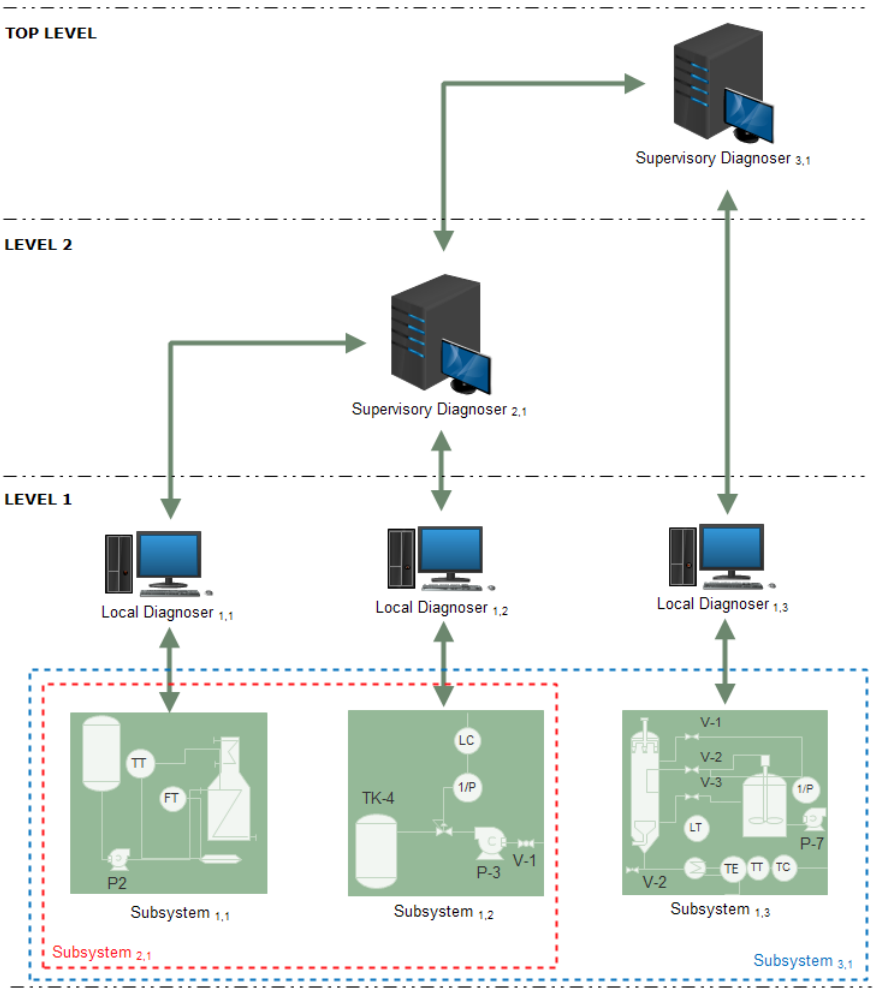


Figure 2.2: Illustration of a decentralized diagnosis architecture.

2.3 Related Work on Decentralized and Distributed Fault Diagnosis

Decentralized and distributed fault diagnosis methods have recently received considerable attention to deal with geographically distributed systems or with systems that may be too large to be diagnosed by one centralized site. An important advantage is that the decentralized and the distributed solutions allow proper separation of industrial knowledge provided that input and outputs are clearly defined. Also, decomposition has been recognized as an important leverage to manage architectural complexity of the systems to be diagnosed.

Researchers have developed several decentralized or distributed diagnosis schemes in the past, mostly in the discrete event framework [Debouk 2000, Pencolé 2005, Wang 2007, Cordier 2007]. Decentralized diagnosis methods have been proposed only recently for continuous or hybrid systems. [Pencolé 2005] presents a method to provide efficient online diagnosis to detect and isolate faults

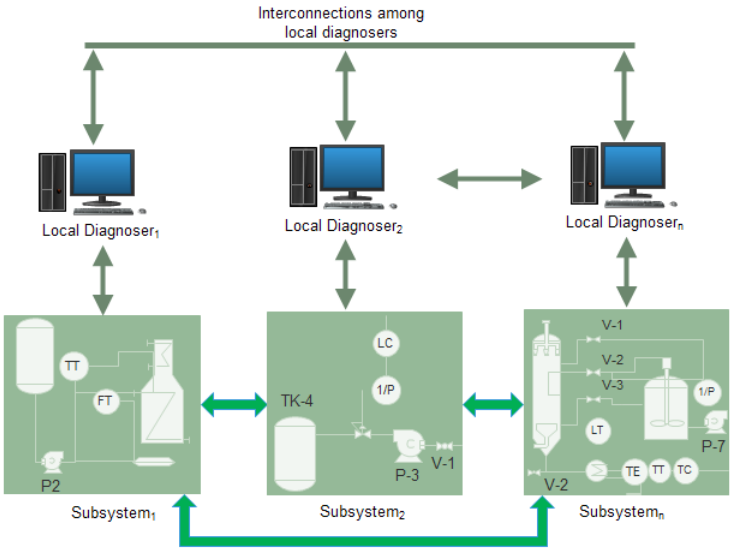


Figure 2.3: Illustration of a distributed diagnosis architecture.

in large discrete event systems. This approach uses a decentralized model of the system and does not require a global model computation. [Wang 2007] proposes a hierarchical framework for discrete event systems using architectures where local sites can issue several types of diagnosis decisions about the presence or absence of each fault including some conditional decisions.

[Zhang 2010] presents a decentralized fault detection scheme where a fault detection estimator is designed for each local subsystem by using local measurements and some communicated information directly from interconnected subsystems. However this approach is limited to a specific class of large-scale nonlinear systems such as nonlinear uncertain systems under certain assumptions. In [Ferdowsi 2012], a decentralized fault diagnosis and prognosis (FDP) methodology is proposed for large-scale systems by using local fault detectors (LFD) or observers for each subsystem based on the measured local states of the subsystem alone for nonlinear continuous-time systems. The disadvantage of this approach is that it requires upper bounds on modeling uncertainties and interconnection functions to be known in nominal operating conditions to construct detection thresholds.

In [Sauter 2006], an isolation filter together with a fault compensation mechanism are proposed for fault detection and isolation (FDI) addressed in a networked framework then a decentralized architecture is proposed using structural analysis. However, this approach does not consider predefined subsystems or constraints and assumes the full availability of the model for distribute a system into subsystems such that the problem of autonomous FDI is solvable for each of these subsystems.

[Cordier 2007] presents a decentralized computation of the diagnosis to avoid the state-explosion problem that appears when dealing with large systems. This decentralized representation relies on two independence properties: state and transition-independence, which are important to get a tractable representation of diagnosis in

the context of both decentralized and incremental approaches.

[Chanthery 2016] proposes a decentralized residual generation using the structural analysis approach with the notion of isolation on request: if local diagnosis is not sufficient to achieve detection and isolation, then the so called hierarchical diagnosers redefine the diagnostic system. The proposal presented in the next chapter of this thesis has a very close relationship with this approach, however it considers the use of Fault-Driven Minimal Structurally Overdetermined (FMSO) sets instead of the whole set of Minimal Structurally Overdetermined (MSO) sets.

Distributed schemes, e.g., [Su 2005, Bregon 2014, Khorasgani 2015], unlike decentralized schemes, do not make use of the upper supervisory levels; instead, they usually use local diagnoser (LD) that communicate their diagnosis results to each other to obtain the global solution.

[Su 2005] presents a general framework for distributed diagnosis for discrete event systems. This framework proposes modelling each local component as a language and modelling the interaction between each pair of components by strings from the set of their shared events.

Distributed diagnosis methods have been proposed recently for continuous systems. [Bregon 2014] present a distributed diagnosis framework for physical systems with continuous behavior using structural model decomposition, using Possible Conflicts approach. They decompose the global system model into submodels that contain sufficient analytical redundancy to perform fault detection. However this is done ignoring pre-existing constraints that may be functional, geographical or privacy-based. We consider pre-existing constraints mandatory and therefore, possible predefined subsystems. [Khorasgani 2015] presents a distributed structural approach to the problem of fault detection and isolation using an algorithm that accepts a just determined subsystem and a set of measurement candidates. It provides a set of diagnosers that are as local as possible by extending local models with their neighboring subsystems models until maximal isolability is achieved.

Decentralized or distributed diagnosis approaches, typically start with a global system model to generate the set of local diagnosers (LD) among which the diagnosis computations get distributed. In Distributed diagnosis approaches each Local Diagnoser makes their diagnosis decision based on only a subset of observable events, and they communicate these decisions to other Local Diagnoser or to a centralized coordinator (in decentralized diagnosis approach case), which uses the global model to generate globally consistent diagnosis solutions. The level of coordination required between the Local Diagnosers depends on how each Local Diagnoser is designed.

Our work considers the same motivation for continuous systems: as complex continuous systems include a large number of components, it is quite unrealistic to rely on a global model of the system. The following section aims at describing important common notions and properties for devellopping our decentralized and distributed fault diagnosis approach.

2.4 Subsystems and Related Notions

This section defines the notion of subsystems and reconsiders the concept of FMSO set in the decentralized and distributed case.

2.4.1 Definitions

In the following, the global level refers to no decentralization and, without loss of generality, we consider two hierarchical levels, the so-called local level and hierarchical level.

Let us consider the system Σ and define the following:

Definition 18 (Global FMSO set)**.** *A global FMSO set is an FMSO set of* $\Sigma(z, x, f)$*. The set of global FMSO sets is denoted by* Φ*.*

In the decentralized case, the decomposition of the system Σ into several subsystems Σ_i is defined as a hierarchical organization of its equations on several levels as shown in Figure 2.4. The equations contained in the set $\Pi_{j,i}$ are equations that are only available at the j^{th} level, because of specific constraints, e.g. confidentiality, distance or difficult access, and therefore not available at the $(j-1)^{th}$ level [Pérez 2015]. Figure 2.4 illustrates an example of decentralized architecture, each square with a dotted line corresponding to a subsystem.

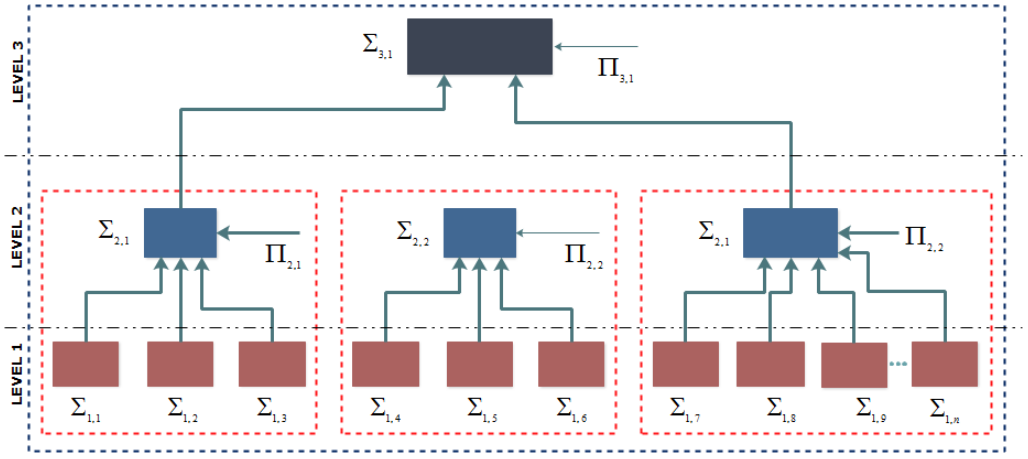


Figure 2.4: Example of a decentralized diagnosis architecture.

In the distributed case, a decomposition of the system Σ into several subsystems Σ_i is defined as a partition of its equations. Let $\Sigma = \{\Sigma_1, \Sigma_2, ..., \Sigma_n\}$ with $\Sigma_i \subseteq \Sigma$, $\bigcup_{i=1}^{n} \Sigma_i = \Sigma$, $\Sigma_i \neq \varnothing$ and $\Sigma_i \cap \Sigma_j = \varnothing$ if $i \neq j$.

Without loss of generality, these two decomposition types lead to n subsystems denoted $\Sigma_i(z_i, x_i, f_i)$, with $i = 1, ..., n$, where z_i is the vector of known variables in Σ_i, x_i the vector of unknown variables in Σ_i and f_i refers to the vector of faults in Σ_i. The set of unknown and known variables and faults of the ith subsystem Σ_i, denoted as X_i, Z_i, and F_i are defined as subsets of variables of X, Z, and F respectively, that are involved in the subsystem Σ_i.

Definition 19 (Local variables)**.** *The set of local variables of the i^{th} subsystem, denoted X_i^l, is defined as the subset of variables of X_i that are only involved in the subsystem Σ_i:*

$$X_i^l = X_i \backslash (\bigcup_{j=1, j\neq i}^{n} (X_i \cap X_j)) \tag{2.1}$$

Definition 20 (Shared Variables)**.** *The set of shared variables of the i^{th} subsystem, denoted as X_i^s, is defined as:*

$$X_i^s = \bigcup_{j=1, j\neq i}^{n} (X_i \cap X_j) = X_i \backslash X_i^l \tag{2.2}$$

The set of shared variables of the whole system Σ is denoted by X^s.

Without loss of generality, it is considered that all known variables of Z_i are local to the subsystem Σ_i, for $i = 1, \ldots, n$. If the same input was applied to several subsystems, it could be artificially replicated.

As an example, we use the two coupled tank system shown in Figure 1.5 and presented in chapter 1 to illustrate the main concepts throughout this chapter.

2.4.2 Example of Decentralized Architecture

Figure 2.5 illustrates the decomposition of the two tank system into a two levels decentralized architecture.

We consider two subsystems of level 1, $\Sigma_{1,1}$ and $\Sigma_{1,2}$. Equations $e_1, e_3, e_6, e_8, e_{10}, e_{11}$ belong to subsystem $\Sigma_{1,1}$. Equations e_2, e_5, e_9 belong to subsystem $\Sigma_{1,2}$.

Subsystem $\Sigma_{2,1}$ includes equations of $\Sigma_{1,1}$ and $\Sigma_{1,2}$ and has a set $\Pi_{2,1} = \{e_4, e_7\}$ that contains equations only available at the second level of the hierarchy.

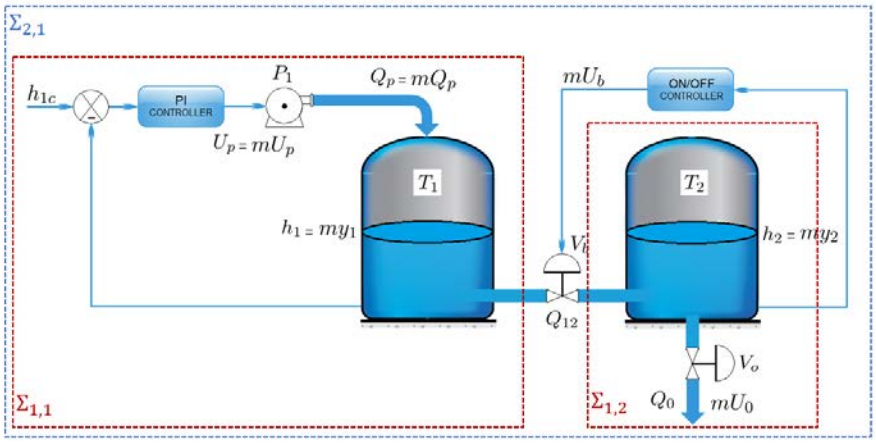


Figure 2.5: Two coupled tanks benchmark in the decentralized case.

The structural representation of this decentralized system is given in Table 2.1.

Equation	Unknown						Faults	
	h_1	h_2	Q_p	Q_0	Q_{12}	U_p	f_1	f_2
e_1	X		X		X		X	
e_3			X			X		
e_6	X					X		
e_8	X							
e_{10}			X					
e_{11}						X		
e_2		X		X	X			X
e_5		X		X				
e_9		X						
e_4	X	X			X			
e_7		X						

Table 2.1: Structural representation of the two coupled tanks benchmark with a two levels decentralized hierarchy.

For the example of the two tanks in the distributed case, $X^l_{1,1} = \{U_p, Q_p\}$: the variables of this subset are only involved in equations of $\Sigma_{1,1}$. $X^l_{1,2} = \{Q_0\}$: it is the only variable only involved in equations of $\Sigma_{1,2}$.

2.4.3 Example of Distributed Architecture

Figure 2.6 illustrates the decomposition of the two tank system into a distributed architecture.

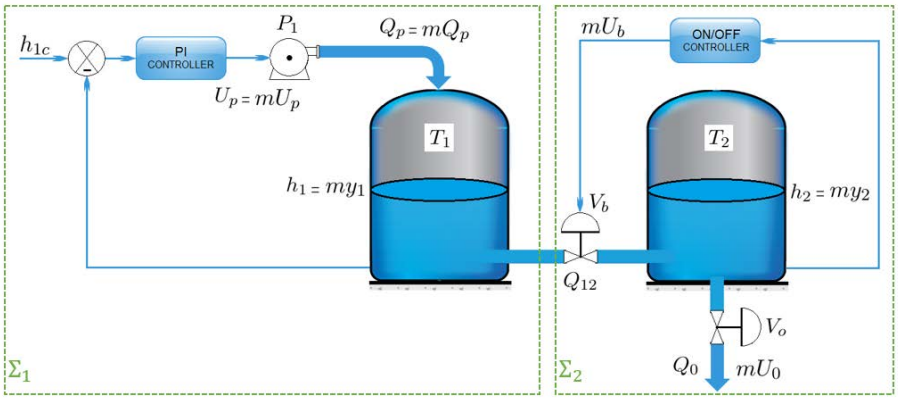


Figure 2.6: Two coupled tanks benchmark in the distributed case.

We consider the system Σ as the union of two subsystems: $\Sigma_1 = \{e_1, e_3, e_6, e_8, e_{10}, e_{11}\}$ and $\Sigma_2 = \{e_2, e_4, e_5, e_7, e_9\}$, each one composed by a tank and its neighbor elements. The structural representation of this system is given in Table 2.2.

Equation	Unknown						Faults	
	h_1	h_2	Q_p	Q_0	Q_{12}	U_p	f_1	f_2
e_1	X		X		X		X	
e_3			X			X		
e_6	X					X		
e_8	X							
e_{10}			X					
e_{11}						X		
e_2		X		X	X			X
e_4	X	X			X			
e_5		X		X				
e_7		X						
e_9		X						

Table 2.2: Structural representation of the two coupled tanks benchmark with two subsystems.

For the example of the two tanks in the distributed case, $X_1^l = \{U_p, Q_p\}$: the variables of this subset are only involved in equations of Σ_1. $X_2^l = \{h_2, Q_0\}$: the variables of this subset are only involved in equations of Σ_2. Then, $X_1^s = X_2^s = \{h_1, Q_{12}\}$.

2.5 Decentralized and Distributed FMSO Sets

This section is based on the concept of Fault-Driven Minimal Structurally Overdetermined (FMSO) set proposed in Chapter 1 and defines special types of FMSO sets that are common to both decentralized and distributed diagnostic design.

Definition 21 (Local FMSO set)**.** *φ is a local FMSO set of $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ if φ is an FMSO set of $\Sigma(z, x, \boldsymbol{f})$ and if $\varphi \subseteq \Sigma_i$, $X_\varphi \subseteq X_i$ and $Z_\varphi \subseteq Z_i^l$. The set of local FMSO sets of Σ_i is denoted by Φ_i^l. The set of all local FMSO sets is denoted by $\Phi^l = \bigcup_{i=1}^{n} \Phi_i^l$.*

Obviously, a local FMSO set for any subsystem Σ_i is also an FMSO set of Σ, hence a global FMSO set.

For the two-tank example, there are three local FMSO sets, $\varphi_1 = \{e_1, e_{10}, \}$, $\varphi_2 = \{e_1, e_3, e_{11}, \}$ and $\{e_1, e_3, e_6, \}$ obtained for Σ_1. These FMSO sets include local and shared variables of Σ_1 and only involve the fault f_1. It can be deduced that to achieve detectability of fault f_1, only the equations included in Σ_1 are required and so that shared variables can be found without the help of equations in other subsystems.

We now define *shared FMSO sets* for a subsystem Σ_i by considering shared variables as known variables and computing FMSO sets. FMSO sets including equations with shared variables are called *shared FMSO sets*.

Definition 22 (Shared FMSO set)**.** *φ is a shared FMSO set of subsystem $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ if φ is an FMSO set of $\tilde{\Sigma}_i(\tilde{z}_i, \tilde{x}_i, \tilde{\boldsymbol{f}}_i)$, where $\tilde{z}_i$ is the vector of variables in $\tilde{Z}_i = Z_i \cup X_i^s$, $\tilde{x}_i$ is the vector of variables in $\tilde{X}_i = X_i^l$, and $\tilde{\boldsymbol{f}}_i = \boldsymbol{f}_i$). The set of shared FMSO sets for Σ_i is denoted by Φ_i^s. The set of all shared FMSO sets is denoted by $\Phi^s = \bigcup_{i=1}^{n} \Phi_i^s$.*

From the above definition, a shared FMSO set φ for subsystem $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ is such that $\varphi \subseteq \Sigma_i$, $X_\varphi \subseteq X_i^l$, $Z_\varphi \cap X_i^s \neq \emptyset$, and $Z_\varphi \subseteq (Z_i \cup X_i^s)$.

Definitions 21 and 22 can also be applied to CMSO sets to define *local CMSO sets* Ψ_i^l and *shared CMSO sets* Ψ_i^s. The set of all shared CMSO sets is denoted by Ψ^s.

Definition 23 (Local CMSO set)**.** *ψ is a local CMSO set of $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ if ψ is an CMSO set of $\Sigma(z, x, \boldsymbol{f})$ and if $\psi \subseteq \Sigma_i$, $X_\psi \subseteq X_i$ and $Z_\psi \subseteq Z_i^l$. The set of local CMSO sets of Σ_i is denoted by Ψ_i^l. The set of all local CMSO sets is denoted by $\Psi^l = \bigcup_{i=1}^{n} \Psi_i^l$.*

Definition 24 (Shared CMSO set)**.** *ψ is a shared CMSO set of subsystem $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ if ψ is an CMSO set of $\tilde{\Sigma}_i(\tilde{z}_i, \tilde{x}_i, \tilde{\boldsymbol{f}}_i)$, where $\tilde{z}_i$ is the vector of variables in $\tilde{Z}_i = Z_i \cup X_i^s$, $\tilde{x}_i$ is the vector of variables in $\tilde{X}_i = X_i^l$, and $\tilde{\boldsymbol{f}}_i = \boldsymbol{f}_i$). The set of shared CMSO sets for Σ_i is denoted by Ψ_i^s. The set of all shared CMSO sets is denoted by $\Psi^s = \bigcup_{i=1}^{n} \Psi_i^s$.*

2.6 FMSO Sets in Distributed Architectures

This section introduces the concept of compound FMSO set and important properties of Fault-Driven Minimal Structurally Overdetermined (FMSO) sets that allow to establish the relation between FMSO sets for the subsystems and FMSO sets for the global system in distributed architectures. These properties are key to demonstrate that whole set of global FMSO sets Φ can be obtained from the set of locally computed FMSO sets, hence achieving a truly distributed architecture.

Definition 25 (Compound FMSO set)**.** *A global FMSO set φ that includes at least one shared FMSO set $\varphi' \in \Phi_i^s$ is called a compound FMSO set. The set of compound FMSO sets of Σ_i is denoted by Φ_i^c. The set of all compound FMSO sets is denoted by $\Phi^c = \bigcup_{i=1}^{n} \Phi_i^c$.*

Definition 26 (Root FMSO set)**.** *If a compound FMSO set $\varphi \in \Phi^c$ includes a shared FMSO set $\varphi' \in \Phi^s$, then φ' is a root FMSO set of φ.*

Definition 27 (Locally detectable fault)**.** *$f \in F_i$ is locally detectable in the subsystem $\Sigma_i(z_i, x_i, \boldsymbol{f}_i)$ if there is an FMSO set $\varphi \in \Phi_i^l$ such that $f \in F_\varphi$.*

Definition 28 (Locally isolable fault)**.** *Given two locally detectable faults f_j and f_k of F_i, $j \neq k$, f_j is locally isolable from f_k if there exists an FMSO set $\varphi \in \Phi_i^l$ such that $f_j \in F_\varphi$ and $f_k \notin F_\varphi$.*

Property 1. *A compound FMSO set φ contains equations from at least two subsystems.*

Property 2. *A local FMSO set $\varphi \in \Phi^l$ is also a global FMSO set.*

Property 3. *A global FMSO set $\varphi \in \Phi$ for which $\exists! i \in 1, \ldots, n$ s.t. $X_\varphi \subseteq X_i^l$ is also a local FMSO set of Σ_i.*

In the following, we show that global FMSO sets can be obtained from locally computed FMSO sets only, by forming compound FMSO sets with shared FMSO sets and shared CMSO sets.

Begin with a simple reasoning. Consider a shared FMSO set $\varphi \in \Phi_i^s$. The particularity of shared FMSO sets is that they are computed hypothesizing that the shared variables they include are known (cf. Definition 22). Actually, this hypothesis is just a trick that allows us to account locally for the FMSO sets that can possibly be generated if equations of other subsystems, indicated by the shared variables, are introduced. However, shared variables are actually unknown so we can define $X_\varphi^s = Z_\varphi \cap X^s$. The shared FMSO set φ can give rise to a global FMSO set if it can be supplemented with sets of equations from other subsystems (more precisely shared FMSO or CMSO sets) to balance the number of shared variables X_φ^s of φ and achieve structural redundancy 1. Let us notice that φ has a structural redundancy of $1-|X_\varphi^s|$. As a matter of fact, every shared variable $x^s \in X_\varphi^s$ decreases the structural redundancy of φ by 1. Consider a shared FMSO set $\varphi' \in \Phi_j^s, j \neq i$ for which x^s is also a shared variable, i.e. $x^s \in X_{\varphi'}^s$. By Proposition 1.6.1, unioning φ' to φ potentially balances the structural redundancy deficiency for one shared variable, say x^s, in φ. However, if φ' introduces new shared variables, these also need to be balanced, each by an additional shared FMSO set. In addition, if x^s is not the only shared variable of φ, the other shared variables each require unioning a different shared FMSO set. The same reasoning also holds if φ' is a shared CMSO set. This leads to the following proposition.

Proposition 2.6.1. *Let $G(\mathbb{X}, \Gamma)$ be a bipartite graph such that $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$ where:*

- *$\mathbb{X}_1 = \Phi^s \cup \Psi^s$ is the set of shared FMSO sets and shared CMSO sets of the system,*
- *$\mathbb{X}_2 = X^s$ is the set of shared variables of the system,*
- *$\Gamma : \mathbb{X}_1 \longrightarrow 2^{\mathbb{X}_2}$ is a function that gives the set of successors of each $\varphi \in \mathbb{X}_1$.*

Let $\varphi \in \mathbb{X}_1$ and $x \in \mathbb{X}_2$ then (φ, x) belongs to the edges of G if $x \in X_\varphi$.

A compound FMSO set $\mathbb{X}_1'$ is built by a subgraph $G_s(\mathbb{X}', \Gamma')$ of $G(\mathbb{X}, \Gamma)$, where $\mathbb{X}' = \mathbb{X}_1' \cup \mathbb{X}_2'$, $\mathbb{X}_1' \subset \mathbb{X}_1$, $\mathbb{X}_2' \subset \mathbb{X}_2$ if:

(i) $G_s(\mathbb{X}', \Gamma')$ *contains no cycles.*

(ii) $\forall \varphi \in \mathbb{X}'_1, \Gamma(\varphi) \subset \mathbb{X}'_2$ *and* $\forall x \in \mathbb{X}'_2 \ \exists \varphi \in \mathbb{X}'_1$ *such that* $\Gamma'(\varphi) = x$.

(iii) The terminal nodes of the graph belong to $\mathbb{X}'_1$.

Proposition 2.6.1 states the conditions for which a union of shared FMSO/CMOS sets originating from different subsystems forms a compound FMSO set. Condition (ii) guarantees that if an FMSO set belongs to the subgraph, then all shared variables are in this subgraph and for all shared variables there exists one shared FMSO/CMSO set that belongs to a subsystem different from any subsystem at the above level. Conditions (i) and (iii) guarantee that the structural redundancy of $\mathbb{X}'_1$ is equal to one and that $\mathbb{X}'_1 = \varphi^c$ is a compound FMSO set.

Equivalently to Proposition 2.6.1 and in accordance with [Chanthery 2016] (Proposition 1 and its proof), compound FMSO can be characterized as sets of FMSO/CMSO that are MSOs with respect to shared variables. Compound FMSO sets can hence be found by running the FMSO generation algorithm (the algorithm run for every subsystem) considering FMSO/CMSO sets as equations and shared variables as unknown variables. Proposition 2.6.1 is stated in a form that makes the optimization problem aiming at only generating the compound FMSO sets that guarantee maximal diagnosability while minimizing shared information easier to formulate as a search problem.

Lemma 1. *The subgraph* $G_s(\mathbb{X}', \Gamma')$ *corresponding to a compound FMSO set has a specific AND/OR tree structure as shown in Figure 2.7.*

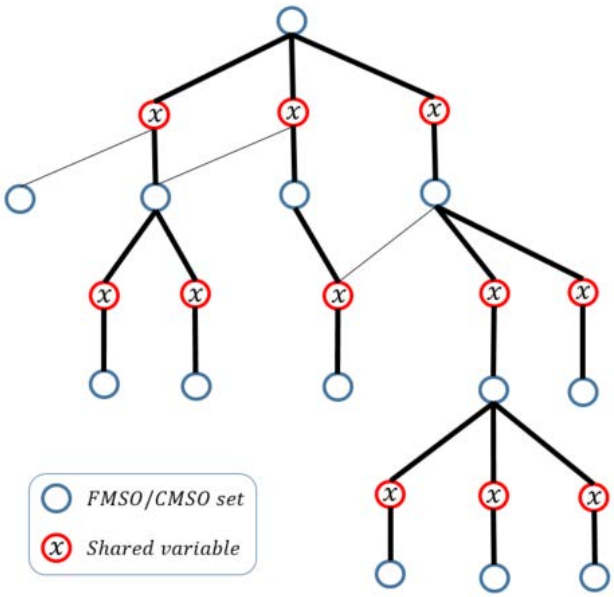


Figure 2.7: AND/OR tree structure of a compound FMSO set.

The FMSO set at the top of Figure 2.7 is considered as the root FMSO set. Its set of shared variables is then included in the structure. For each of them, only

one FMSO set is chosen among the FMSO/CMSO sets that include the shared variable. For each chosen FMSO/CMSO set, the shared variables are included in the structure. This property repeats down the graph levels until there is no additional shared variable to include in the structure. We talk of an *iterative matching procedure*. It can be proved that all the global FMSO sets can be obtained from locally computed FMSO sets.

Proposition 2.6.2. *The set of global FMSO sets* Φ *is given by the union of the set of local FMSO sets* Φ^l *and the set of compound FMSO sets* Φ^c*.*

$$\Phi = \Phi^l \cup \Phi^c \tag{2.3}$$

2.7 Conclusion

This chapter presents the different architectural options for fault diagnosis. Centralized diagnosis architecture is simple but there exist some contexts where its implementation will become impossible in practice. Decentralized and distributed diagnosis architectures are then presented. The main difference between decentralized and distributed architectures is the existence of supervisory levels in the case of decentralized architectures. Important notions, FMSO sets and properties for decentralized and distributed fault diagnosis systems are then presented and illustrated on a two tank benchmark. An important proposition states the conditions for which a union of shared FMSO/CMSO sets originating from different subsystems forms a global FMSO set.

The following chapters present the operational procedures for the implementation of decentralized diagnostic systems and distributed diagnostic systems based on the concepts presented in this chapter.

Part II

Decentralized and Distributed Fault Diagnosis Algorithms

Chapter 3

Decentralized Diagnosis via Structural Analysis

Contents

3.1 Introduction

Centralized fault diagnosis architectures are sometimes not applicable practical for large-scale interconnected systems such as distribution systems, telecommunication networks, water distribution networks, fluid power systems. Actually this type of systems require sensing, processing and transmission of a large number of variables measured from various parts of the system. Centralized architectures can be very expensive and lack robustness for such large-scale interconnected systems. Thereby, decentralized control of interconnected systems has been deeply analyzed in the litterature [Corfmat 1976, Siljak 2011, Wang 2017] while decentralized fault diagnosis is being dealt only recently.

On the other hand, a decentralized diagnosis architecture is interesting from a design point of view. The integration of fault management functionality design and development into the development processes of the corresponding system's functions is expected to be more efficient than the traditional approach. Subsystem development teams and processes are traditionally organized for the nominal functions they develop which are then integrated together. Decentralized diagnosis architectures adapt to this traditional organization and can be easily merged into the processes, designing a diagnoser for each different function. Such function-failure codesign involves considering fault diagnosis functionality as an integral component of the system from the early stages of design.

The diagnoser architecture developed in this chapter is composed of local diagnosers which work with local models of their subsystems as seen in Figure 3.1. Diagnosis ambiguity among local diagnosers is resolved by a supervisory diagnoser at a higher level. This architecture is a natural match to the systems engineering process which proceeds with a functional decomposition of a system into subsystems. The architecture is hierarchically scalable, and implements diagnosis based on ARRs. The approach to ARR generation and implementation adopted in the architecture is presented.

A distributed architecture as understood in this thesis assumes every diagnosis unit to be identical in terms of role, with communication possible between any two diagnosis nodes. The decentralized architecture presented in this chapter is composed of local diagnosers whose results are coordinated by a supervisory diagnoser. There is no intra-level communication in the architecture i.e local diagnosers at a level do not communicate with each other. Inter-level communication between local diagnosers and the supervisory diagnoser of their level serves to disambiguate diagnosis results. In the thesis such an architecture with dissimilar roles of diagnosis units is understood as implementing decentralization. A similar diagnoser architecture to the one developed here is presented in [Console 2007] for systems modeled in the qualitative framework.

This work resumes the work of [Chanthery 2016] and it must be considered as a direct continuation. The proposed algorithm has the same goal but it gains efficiency over the algorithm of [Chanthery 2016] by taking benefit of the concept of FMSO set introduced in the previous chapter. In a similar way, the level of diagnosability deployed can be dynamically changed depending upon fault management requirements. This feature is known as *isolation on request* [Chanthery 2016]. The algorithm also inherits the property that was demonstrated in [Chanthery 2016] that assesses that decentralized diagnosis with ARRs has equivalent diagnosability properties as a centralized diagnoser.

Let us notice that the diagnosis process consists of two stages: offline and online. The offline stage determines analytical residual generators for which a structural analysis based algorithm is presented and the online stage determines the fault with a hierarchical and local residual generator bank as well as a fault signature matrix.

3.2 An Algorithm for Decentralized Diagnoser Design

The diagnoser design is done *offline* and is implemented by developing the steps of the algorithm 3.1. These steps are performed for each subsystem $\Sigma_{j,i}$ $i = 1, \ldots, n_j$ at each level $j = 1, \ldots, m$, with a nested loop. Here j is the level in the hierarchy, and i the enumeration of subsystems at each level.

If the faults are not detectable or isolable at their corresponding level, diagnosers are developed at the higher levels until the diagnosability objective is achieved. The diagnosis hierarchy is constrained by the possible inter-level communication, which is defined by a set of bipartite graphs.

Algorithm 3.1: Decentralized Diagnoser Design.

1 $n_0 = 1$, $E_{0,1} = \emptyset$;
  /* From level 1 to level m do: */
2 **for** $j \leftarrow 1$ **to** m **do**
3 $\Delta_{j,i} \leftarrow$ Compute subsystems corresponding to level $j-1$ that have links to subsystem i at level j.;
  /* From subsystem 1 to subsystem n_j do: */
4 **for** $i \leftarrow 1$ **to** n_j **do**
5 $\Pi_{j,i} \leftarrow$ Load additional equations for system $\Sigma_{j,i}$;
6 $\Sigma_{j,i} = \Pi_{j,i} \cup (\bigcup_{i \in \Delta_{j,i}} E_{j-1,i})$;
7 $\Phi^l_{j,i} \leftarrow$ Compute local FMSO sets of $\Sigma_{j,i}$;
  /* Compute ARRs of $\Sigma_{j,i}$ from local FMSO sets */
8 $ARR_{j,i} \leftarrow$ Compute analytical residual generators of $\Phi^h_{j,i}$;
9 **if** *there is any fault* $f \in F_{j,i}$ *not locally detectable or not locally isolable with the set of local FMSO sets* $\Phi^l_{j,i}$ **then**
10 $\Phi^s_{j,i} \leftarrow$ Compute shared FMSO sets of $\Sigma_{j,i}$;
11 $\Psi^s_{j,i} \leftarrow$ Compute shared CMSO sets of $\Sigma_{j,i}$;
12 $E_{j,i} = \{e \in \Sigma / \exists S_{j,i} \in \Phi^s_{j,i} \cup \Psi^s_{j,i} \wedge e \in S_{j,i}\}$;

Definition 29 (Inter-level communication)**.** *The inter-level communication is represented by a set of* $m-1$ *bipartite graphs* $S^j_{j-1}(\mathbb{N}^j_{j-1}, \mathbb{L}^j_{j-1})$, $j = 2, \ldots, m$. $S^j_{j-1}(\mathbb{N}^j_{j-1}, \mathbb{L}^j_{j-1})$ *is a bipartite graph such that* $\mathbb{N}^j_{j-1} = \mathbb{N}_{j-1} \cup \mathbb{N}_j$, *where:*

- $\mathbb{N}_{j-1} = \{n_{j-1,i}, i = 1, \ldots, n_{j-1}\}$ *is a set of nodes corresponding to the subsystems* $\Sigma_{j-1,i}, i = 1, \ldots, n_{j-1}$, *of level* $j-1$,
- $\mathbb{N}_j = \{n_{j,i}, i = 1, \ldots, n_j\}$ *is a set of nodes corresponding to the subsystems* $\Sigma_{j,i}, i = 1, \ldots, n_j$, *of level* j,
- $\mathbb{L}^j_{j-1} = \{l_{\nu,\xi}, \nu = 1, \ldots, n_{j-1}, \xi = 1, \ldots, n_j\}$ *is a set of edges such that the edge* $l_{\nu,\xi}$ *between node* $n_\nu \in \mathbb{N}_{j-1}$ *and* $n_\xi \in \mathbb{N}_j$ *exists if communication is possible between subsystem* $\Sigma_{j-1,\nu}$ *at level* $j-1$ *and subsystem* $\Sigma_{j,\xi}$ *at level* j.

In Algorithm 3.1, the inter-level communication is taken into account by $\Delta_{j,i}$, where $\Delta_{j,i} = \{\Sigma_{j-1,\nu} / l_{\nu,i}$ exists in $S^j_{j-1}(\mathbb{N}^j_{j-1}, \mathbb{L}^j_{j-1})\}$, for $j = 2, \ldots, m$, and $\Delta_{1,i} = \emptyset$. In other words, for $j = 2, \ldots, m$ representing the level, $\Delta_{j,i}$ contains all subsystems of level $j-1$ that have connection with subsystem $\Sigma_{j,i}$, i.e. subsystem i at level j.

The equations contained in the set $\Pi_{j,i}$ in Algorithm 3.1 (Line 5) are additional equations that are only available at level j for forming subsystem $\Sigma_{j,i}$. The restriction on these equations may originate from different constraint types, e.g. confidentiality, distance and difficult access and they are not therefore available at level $j-1$.

The equations contained in the set $E_{j,i}$ in Algorithm 3.1 (Line 11) are all the equations included in the shared FMSO and CMSO of system $\Sigma_{j,i}$.

At line 1, n_0 and $E_{0,1}$ are initial conditions for a virtual level 0 that allow us to solve the recursive equations for the following levels.

Line 2 is a **for** instruction on the levels from 1 to m. At line 3, the set $\Delta_{j,i}$ is determined as defined above.

Then a loop is started that covers all subsystems from $i = 1$ to $i = n_j$ of level j. $\Pi_{j,i}$ is loaded at line 5 and the subsystem $\Sigma_{j,i}$ is formed at line 6 from the additional equations $\Pi_{j,i}$ and the union of equations coming from the shared FMSO and CMSO sets of the lower level, i.e. $E_{j-1,i}$.

Local FMSO sets for $\Sigma_{j,i}$ are computed at line 7 and the set of ARRS for subsystem $\Sigma_{j,i}$ is computed at line 8. Shared FMSO and CMSO sets are computed at lines 10 and 11 only if the faults of the "children" subsystems of level 0 are not isolable at this level. These latter equations are to be sent to the next level.

The algorithm hence computes recursively, by developing the necessary levels, all the analytical residual generators that guarantee to isolate all the faults.

Figure 3.1 illustrates the architecture of the proposed algorithm for three levels ($m = 3$), each square with a dotted lines corresponding to a subsystem.

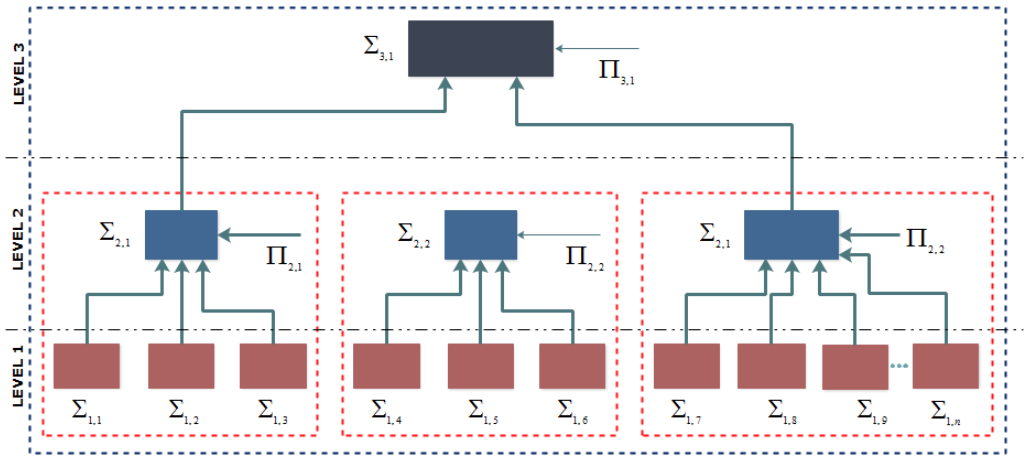


Figure 3.1: Architecture of the decentralized diagnoser designed *offline*.

3.3 Online Implementation of the Decentralized Diagnoser

After the **offline** decentralized diagnoser design using Algorithm 3.1, the **online** implementation of the fault diagnosis system requires some additional elements.

Definition 30 (Fault Signature Matrix of a subsystem)**.** *Given a set $ARR_{j,i}$ composed of $n^r_{j,i}$ ARRs and $F_{j,i}$ the set of considered $n^f_{j,i}$ faults for the subsystem $\Sigma_{j,i}$, the signature of a fault $f \in F_{j,i}$ is the binary vector $FS_{j,i}(f) = [\tau_1, \tau_2, ...\tau_{n^r_{j,i}}]^T$ where τ_k, $k = 1...n^r_{j,i}$, is computed from $ARR_{j,i} \times F_{j,i} \longrightarrow 0, 1$ so that $\tau_k = 1$, if the*

equation affected by $F_{j,i}$ is involved in $arr_k \in ARR_{j,i}$, otherwise $\tau_k = 0$. The signatures of all the faults in $F_{j,i}$ together constitute the fault signature matrix $FSM_{j,i}$ for subsystem $\Sigma_{j,i}$, i.e. $FSM_{j,i} = [FS_{j,i}(f_1), \ldots, FS_{j,i}(f_{n^f_{j,i}})]^T$.

The diagnoser is implemented **online** as a hierarchical residual generator bank based on the local FMSO sets generated for each subsystem at each level. With the system inputs and outputs, all the computed hierarchical residual generators are used **online** to detect and isolate faults along the levels. The fault isolation process happens for each subsystem for which local FMSO sets have been computed based on its fault signature matrix.

Let us notice that the computations are carried up the levels only if all the faults are not yet isolable. This is the idea of *isolation on request*.

3.4 Application to the Four-Tank System

The decentralized diagnoser is illustrated on the four-tank system described in example 3.4.1 and shown in Figure 3.2 considering the equations of Table 3.1.

Example 3.4.1. *Consider the benchmark problem of four coupled tanks depicted in Figure 3.2, that provides a continuous water flow to consumers.*

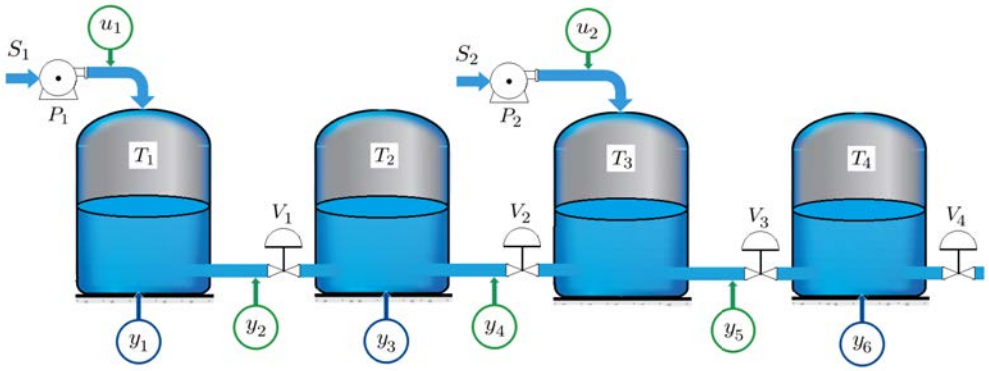


Figure 3.2: Four coupled tanks benchmark from [Khorasgani 2015].

The components are tanks T_1, T_2, T_3 and T_4, pumps P_1 and P_2, proportional valves V_1, V_2, V_3 and V_4, three level sensors with sensoring variables y_1, y_3 and y_6, three flow sensors with sensoring variables y_2, y_4 and y_5 and two flow sensors for the input flow rates u_1 and u_2.

The global model $\Sigma(z, x, \mathtt{f})$ for this system is composed of twenty equations e_1 to e_{20} as shown in Table 3.1, they relate the known variables $Z = \{u_1, u_2, y_1, y_2, y_3, y_4, y_5, y_6\}$, the unknown variables $X = \{\dot{p}_1, p_1, \dot{p}_2, p_2, \dot{p}_3, p_3, \dot{p}_4, p_4, q_{in1}, q_{in2}, q_1, q_2, q_3, q_4\}$ and the set of system faults $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$. All proportional valves are assumed to be fully open.

Now, we consider each tank as a subsystem so this system is decomposed into four subsystems at the bottom level, level 1. Additional equations referring to the subsystems up the levels are the following :

Table 3.1: Equations for the four-tank system.

$e_1:$	$\dot{p}_1 = \frac{1}{C_{T_1} + f_1}(q_{in1} - q_1)$	$e_4:$	$q_{in1} = u_1$
$e_2:$	$q_1 = \frac{p_1 - p_2}{R_{P_1} + f_2}$	$e_5:$	$p_1 = y_1$
$e_3:$	$p_1 = \int \dot{p}_1 dt$	$e_6:$	$q_1 = y_2$
$e_7:$	$\dot{p}_2 = \frac{1}{C_{T_2} + f_3}(q_1 - q_2)$	$e_{10}:$	$p_2 = y_3$
$e_8:$	$q_2 = \frac{p_2 - p_3}{R_{P_2} + f_4}$	$e_{11}:$	$q_2 = y_4$
$e_9:$	$p_2 = \int \dot{p}_2 dt$		
$e_{12}:$	$\dot{p}_3 = \frac{1}{C_{T_3}}(q_{in2} + q_2 - q_3)$	$e_{15}:$	$q_{in2} = u_2$
$e_{13}:$	$q_3 = \frac{p_3 - p_4}{R_{P_3} + f_5}$	$e_{16}:$	$q_3 = y_5$
$e_{14}:$	$p_3 = \int \dot{p}_3 dt$		
$e_{17}:$	$\dot{p}_4 = \frac{1}{C_{T_4} + f_6}(q_3 - q_4)$	$e_{19}:$	$p_4 = \int \dot{p}_4 dt$
$e_{18}:$	$q_4 = \frac{p_4}{R_{P_4}}$	$e_{20}:$	$p_4 = y_6$

$$\Pi_{1,1} = \Pi_{1,2} = \Pi_{1,3} = \Pi_{1,4} = \{\emptyset\} \tag{3.1}$$

$$\Pi_{2,1} = \{e_2, e_6\}, \Pi_{2,2} = \{e_{13}, e_{16}\} \tag{3.2}$$

$$\Pi_{3,1} = \{e_8, e_{11}\} \tag{3.3}$$

The hierarchical decomposition of this system is shown in Figure 3.3, where in level 1, the 4 subsystems are each composed of a tank, in level 2 the information of tanks 1 and 2 is grouped in subsystem $\Sigma_{2,1}$ and tanks 2 and 3 in subsystem $\Sigma_{2,2}$, the equations of $\Pi_{2,1}$ and $\Pi_{2,1}$ being respectively added to each. At level 3, the remaining communication between tank 2 and tank 3 is considered to form subsystem $\Sigma_{3,1}$ including the additional equations of $\Pi_{3,1}$.

The models of each subsystem of level 1 are shown in Table 6.2.

3.4.1 Centralized Diagnoser

As a reference, the FMSO sets are determined for the whole system considered globally in order to determine maximal fault isolation.

According to the results of Table 3.3, it can be seen that all faults can be detected and isolated with a centralized diagnoser for the four-tank system.

3.4.2 Decentralized Diagnoser

The algorithm 3.1 for decentralized diagnosis design is now applied. According to the characteristics adopted for this example, we consider three levels $j = 1, 2$ and 3

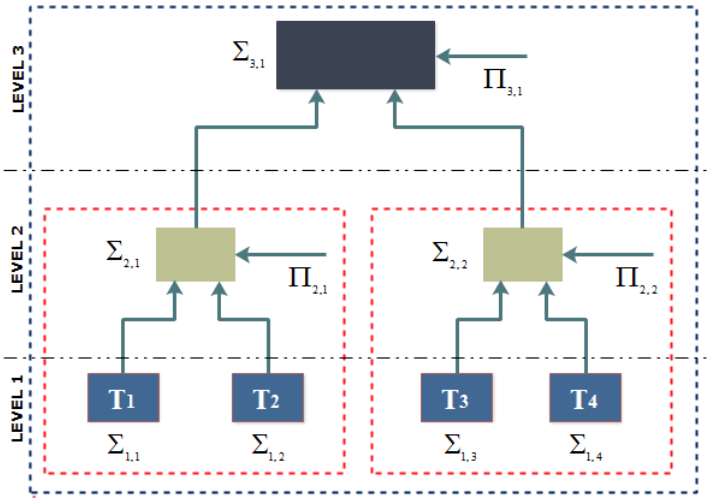


Figure 3.3: Architecture of the decentralized diagnoser designed for the four-tank system.

Table 3.2: Model decomposition of the four-tank system into subsystems $\Sigma_{1,i}$ $(i = 1, 2, 3, 4)$.

$$\Sigma_{1,1} = \begin{cases} \Sigma_{1,1} = \{e_1, e_3, e_4, e_5\} & F_1 = \{f_1\} \\ X_1 = \{\dot{p_1}, p_1, q_{in1}, q_1\} & Z_1 = \{u_1, y_1\} \end{cases}$$

$$\Sigma_{1,2} = \begin{cases} \Sigma_{1,2} = \{e_7, e_9, e_{10}\} & F_2 = \{f_3\} \\ X_2 = \{\dot{p_2}, p_2, q_1, q_2\} & Z_2 = \{y_3, \} \end{cases}$$

$$\Sigma_{1,3} = \begin{cases} \Sigma_{1,3} = \{e_{12}, e_{14}, e_{15}, \} & F_3 = \{\} \\ X_3 = \{\dot{p_3}, p_3, q_{in2}, q_2, q_3\} & Z_3 = \{u_2\} \end{cases}$$

$$\Sigma_{1,4} = \begin{cases} \Sigma_{1,4} = \{e_{17}, e_{18}, e_{19}, e_{20}\} & F_4 = \{f_6\} \\ X_4 = \{\dot{p_4}, p_4, q_3, q_4\} & Z_4 = \{y_6\} \end{cases}$$

Four-Tank system Global Diagnoser	
Max fault isolability	$[f_1]$, $[f_2]$, $[f_3]$, $[f_4]$, $[f_5]$, $[f_6]$
FMSO sets	165 FMSO sets

Table 3.3: FMSO sets for the Global System.

with four subsystems $i = 1, 2, 3, 4$ for the first level, two subsystems for the second level $(j = 2)$, and one subsystem $i = 1$, for the third level $(j = 3)$.

As a previous step, with the information of Table 3.1, it is possible to determine the vector of shared variables as $X^s = \{q_1, p_2, q_2, p_3, q_3, p_4\}$. These will be used to compute the *shared FMSO sets*.

First, *local FMSO sets* are calculated for each subsystem of level 1.

$$\Phi^l_{1,1} = \Phi^l_{1,2} = \Phi^l_{1,3} = \Phi^l_{1,4} = \{\emptyset\} \tag{3.4}$$

The set of local FMSO sets for $\Sigma_{1,1}$, $\Sigma_{1,2}$, $\Sigma_{1,3}$ and $\Sigma_{1,4}$ are empty. Hence, with

no additional information, no fault can be diagnosed at level 1.

Next, for each one of the 4 subsystems, shared variables X_i^s, $i = 1, 2, 3, 4$ are now assumed to be known and *shared FMSO sets* and *shared CMSO sets* are computed. Results are given in Tables 3.4, 3.5, 3.6 and 3.7.

$\Sigma_{1,1}$	
Max fault isolability	$[f_1]$
Shared FMSO sets $\Phi^s_{1,1} = \{\varphi_1\}$	$\varphi_1 = \{e_1, e_3, e_4, e_5\}$
Shared CMSO sets $\Psi^s_{1,1} = \{\emptyset\}$	

Table 3.4: Subsystem $\Sigma_{1,1}$: $\Phi^s_{1,1}$, $\Psi^s_{1,1}$.

$\Sigma_{1,2}$	
Max fault isolability	$[f_3]$
Shared FMSO sets $\Phi^s_{1,2} = \{\varphi_2\}$	$\varphi_1 = \{e_7, e_9\}$
Shared CMSO sets $\Psi^s_{1,2} = \{\psi_1\}$	$\psi_1 = \{e_{10}\}$

Table 3.5: Subsystem $\Sigma_{1,2}$: $\Phi^s_{1,2}$, $\Psi^s_{1,2}$.

$\Sigma_{1,3}$	
Max fault isolability	$[\emptyset]$
Shared FMSO sets $\Phi^s_{1,3} = \{\emptyset\}$	
Shared CMSO sets $\Psi^s_{1,3} = \{\psi_2\}$	$\psi_2 = \{e_{12}, e_{14}, e_{15}\}$

Table 3.6: Subsystem $\Sigma_{1,3}$: $\Phi^s_{1,3}$, $\Psi^s_{1,3}$.

$\Sigma_{1,4}$	
Max fault isolability	$[f_6]$
Shared FMSO sets $\Phi^s_{1,4} = \{\varphi_3\}$	$\varphi_1 = \{e_{17}, e_{18}, e_{19}\}$
Shared CMSO sets $\Psi^s_{1,4} = \{\psi_3\}$	$\psi_3 = \{e_{20}\}$

Table 3.7: Subsystem $\Sigma_{1,1}$: $\Phi^s_{1,4}$, $\Psi^s_{1,4}$.

Then, according to the algorithm, at level 2, the subsystems $\Sigma_{2,1}$ and $\Sigma_{2,2}$ are considered. For this purpose, shared FMSO and CMSO sets of the children systems of level 1 are considered together with the additional equations of $\Pi_{2,1}$ and $\Pi_{2,2}$ to form $\Sigma_{2,1}$ and $\Sigma_{2,2}$, respectively :

$$\Sigma_{2,1} = \{e_1, e_3, e_4, e_5, e_7, e_9, e_{10}\} \cup \{e_2, e_6\} \tag{3.5}$$

$$\Sigma_{2,2} = \{e_{12}, e_{14}, e_{15}, e_{17}, e_{18}, e_{19}, e_{20}\} \cup \{e_{13}, e_{16}\} \tag{3.6}$$

Local FMSO sets are calculated for each subsystem $\Sigma_{2,1}$ and $\Sigma_{2,2}$ at level 2.

$\Sigma_{2,1}$	
Max fault isolability	$[f_1], [f_2]$
Local FMSO sets $\Phi^l_{1,1} = \{\varphi_4, \varphi_5, \varphi_6, \varphi_7, \}$	$\varphi_4 = \{e_2, e_5, e_6, e_{10}\}$
	$\varphi_5 = \{e_1, e_3, e_4, e_5, e_6\}$
	$\varphi_6 = \{e_1, e_2, e_3, e_4, e_6, e_{10}\}$
	$\varphi_7 = \{e_1, e_2, e_3, e_4, e_5, e_{10}\}$

Table 3.8: Subsystem $\Sigma_{2,1}$: $\Phi^l_{2,1}$.

$\Sigma_{2,2}$	
Max fault isolability	$[f_6]$
Local FMSO sets $\Phi^l_{2,2} = \{\varphi_8\}$	$\varphi_8 = \{e_{16}, e_{17}, e_{18}, e_{19}, e_{20}\}$

Table 3.9: Subsystem $\Sigma_{2,2}$: $\Phi^l_{2,2}$.

As shown in Tables 3.8 and 3.9, it is possible to detect and isolate faults f_1, f_2 and f_6 at level 2. To complete diagnosis, *shared FMSO sets* and *shared CMSO sets* are computed for $\Sigma_{2,1}$ and $\Sigma_{2,2}$ and these are sent to the third level.

$\Sigma_{2,1}$	
Max fault isolability	$[f_1], [f_2]$
Shared FMSO sets $\Phi^s_{2,1} = \{\varphi_9, \varphi_{10}, \varphi_{11}, \varphi_{12}\}$	$\varphi_9 = \{e_7, e_9\}$
	$\varphi_{10} = \{e_2, e_5\}$
	$\varphi_{11} = \{e_1, e_2, e_3, e_4\}$
	$\varphi_{12} = \{e_1, e_3, e_4, e_5\}$
Shared CMSO sets $\Psi^s_{2,1} = \{\psi_4, \psi_5\}$	$\psi_4 = \{e_6\}$
	$\psi_5 = \{e_{10}\}$

Table 3.10: Subsystem $\Sigma_{2,1}$: $\Phi^s_{2,1}$, $\Psi^s_{2,1}$.

$\Sigma_{2,2}$	
Max fault isolability	$[f_5], [f_6]$
Shared FMSO sets $\Phi^s_{2,1} = \{\varphi_{13}, \varphi_{14}\}$	$\varphi_{13} = \{e_{17}, e_{18}, e_{19}\}$
	$\varphi_{14} = \{e_{13}\}$
Shared CMSO sets $\Psi^s_{2,1} = \{\psi_6, \psi_7, \psi_8\}$	$\psi_6 = \{e_{20}\}$
	$\psi_7 = \{e_{16}\}$
	$\psi_8 = \{e_{12}, e_{14}, e_{15}\}$

Table 3.11: Subsystem $\Sigma_{2,1}$: $\Phi^s_{2,1}$, $\Psi_{2,1}$.

Finally, shared FMSO and CMSO sets of $\Sigma_{2,1}$ and $\Sigma_{2,2}$ are put together with the additional equations of $\Pi_{3,1}$ to form $\Sigma_{3,1}$:

$$\Sigma_{3,1} = \{e_1, e_2, ..., e_7, e_9, e_{10}, e_{12}, e_{13}, ..., e_{20}\} \cup \{e_8, e_{11}\} \tag{3.7}$$

With the information within $\Sigma_{3,1}$ the detection and isolation is completed for the six faults of interest with six *local FMSO sets*:

$\Sigma_{3,1}$	
Max fault isolability	$[f_3]$, $[f_4]$
Local FMSO sets selected:	
$\Phi^l_{3,1} = \{\varphi_{15}, \varphi_{16}\}$	
	$\varphi_{15} = \{e_6, e_7, e_9, e_{10}, e_{11}\}$
	$\varphi_{16} = \{e_8, e_{10}, e_{11}, e_{13}, e_{16}, e_{20}\}$

Table 3.12: $\Sigma_{3,1}$: $\Phi^l_{3,1}$.

Based on the found local FMSO sets of $\Sigma_{3,1}$ given in Table 3.12, we conclude that it is possible to detect the six faults of the system with analytical residual generators of level 3. However, one could use analytical residual generators obtained in a recursive way. Local analytical redundancy relations for subsystem $\Sigma_{2,1}$ and $\Sigma_{2,1}$ can be used to isolate f_1 f_2, and f_6 respectively. Then, some of the six analytical redundancy relations computed from the local FMSO sets of $\Sigma_{3,1}$ can complete the isolation. The isolation pattern is shown in the fault signature matrix of the Table 3.13.

	Faults					
	f_1	f_2	f_3	f_4	f_5	f_6
$arr_1 \in ARR_{2,1}$	X					
$arr_2 \in ARR_{2,1}$		X				
$arr_3 \in ARR_{2,2}$						X
$arr_5 \in ARR_{2,2}$					X	
$arr_4 \in ARR_{3,1}$			X			
$arr_6 \in ARR_{3,1}$				X		

Table 3.13: Fault signature matrix issued from the decentralized diagnoser.

3.5 Conclusion

A proposal is presented for the design of decentralized fault diagnosis for systems that have constraints of confidentiality, distance or limited access to some information. An algorithm for decentralized diagnoser design is proposed. It uses the notion of inter-level communication. The algorithm computes recursively, by developing the just-needed levels, all the analytical residual generators that guaranty to isolate all the faults. The online implementation of the diagnoser gives rise to one fault signature matrix per subsystem. Computations are carried up the levels only if all the faults are not yet isolable. This is the idea of isolation on request. The methodology is illustrated on a four-tank system.

Chapter 4

Fault-Driven Structural Diagnosis Approach in a Distributed Context

Contents

4.1 Introduction

Distributed approaches are recommended for large complex systems with constraints such as communication bandwidth or geographic distribution. In some cases, a distributed diagnosis architecture may be the only viable solution given structural, computational and robustness issues.

In large-scale systems, diagnosis algorithms must account for two real-time requirements [Boem 2011]:

1. enough computation power for processing all the necessary measurements,

2. enough communication bandwidth in order to gather all the measurements to the place where they are processed.

In addition to the economic implications related to the first requirement, it should be noted that the second requirement can be even more difficult to achieve if for example the system covers a large geographic area and the measurements are distributed, so that they cannot be directly wired to the processing computer. Moreover, there

are contexts where a centralized architecture, even if feasible, would be undesirable because of several factors including size, robustness and security issues, e.g., aircraft and other transportation systems, large-scale energy or industrial plants, power generation, etc.

While distribution is often dictated by physical constraints, it has several other appealing properties over a centralized approach, including fault tolerance, scalability, and reusability. Fault tolerance stems from the ability of distributed systems to keep operating when one or more components are faulty. The scalability comes from reduced costs of system setup and update, communication, and decision making. Finally, when reconfiguration is required, implying to change some components, actuators, or sensors, it can be easier to modify part of the distributed system impacted by the changes than to overhaul the centralized system as a whole [Grbovic 2012].

This chapter presents the operational procedure for the implementation of a distributed fault diagnosis system considering the definitions of the previous Chapter 2. First, we present an algorithm for the calculation of all global FMSO sets based on local information only and next an algorithm for distributed generation of an optimized set of global FMSO sets. This offline stage is developed for computing analytical residual generators. The online stage achieves fault detection with the previously designed residual generator bank forming local diagnosers (LD) and a fault signature matrix.

4.2 An Algorithm for Distributed Diagnoser Design

4.2.1 Distributed Generation of all Global FMSO Sets

Like [Khorasgani 2015], this approach assumes the non-availability of a global system model and it guaranties maximal diagnosability, i.e. the same diagnosability as a centralized approach. Unlike [Khorasgani 2015], it is proved, as a result of the properties FMSO sets, that it is possible to obtain the set of global FMSO sets without recomputing FMSO sets for the local models extended by neighboring subsystem models. Instead, our approach uses a search algorithm that identifies the sets of shared FMSO/CMSO sets computed locally that can become global FMSO sets. Algorithm 4.1 [Pérez 2016] implements the procedure for computing the set of global FMSO sets following the proposed distributed approach.

Algorithm 4.1 relies on various definitions introduced in Chapter 2 and on Proposition 2.6.1. The notations that are used below are the same as those used in this previous chapter.

A distributed architecture as understood in this thesis assumes every diagnosis unit to be identical in terms of role, with communication possible between any two diagnosis nodes. As opposed to the decentralized diagnosis architecture, there is no decomposition hierarchy for the system. Instead, the system Σ is decomposed into n subsystems $\Sigma_i, i = 1, \ldots, n$ whose associated models define a partition of the equations Σ.

Algorithm 4.1: Generation of the set of global FMSO sets.

```
1  Φ = ∅;
2  for i=1...n do
3      Φ_i^l ← Calculate local FMSO sets of Σ_i;
4      Φ_i^s ←Calculate shared FMSO sets of Σ_i;
5      Ψ_i^s ← Calculate shared CMSO sets of Σ_i;
6      for each shared FMSO set φ ∈ Φ_i^s do
7          Label φ as root FMSO: φ_r ← φ;
8          Let X^s_{φ_r} be the set of shared variables of φ_r;
9          while it is possible to find a set φ^c ⊇ φ_r that can be a set 𝕏'_1 in
             Proposition 2.6.1 and such that φ^c is not included in Φ do
10             Store the global FMSO set φ^c: Φ ← Φ ∪ φ^c;
11     Φ ← Φ ∪ Φ_i^l
12 Return Φ;
```

Consider a root FMSO set φ_r as defined in Definition 26 and let $G(\mathbb{X}, \Gamma)$ be a bipartite graph such that $\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$ where:

- $\mathbb{X}_1 = \Phi^s \cup \Psi^s$ is the set of shared FMSO sets and shared CMSO sets of the system,
- $\mathbb{X}_2 = X^s$ is the set of shared variables of the system,
- $\Gamma : \mathbb{X}_1 \longrightarrow 2^{\mathbb{X}_2}$ is a function that gives the set of successors of each $\varphi \in \mathbb{X}_1$.

The procedure to compute a global FMSO set φ^c, resulting from the set denoted $\mathbb{X}'_1$ in Proposition 2.6.1, starts by searching for a matching that covers each shared variable of the root FMSO set φ_r, i.e. of the set $X^s_{\varphi_r}$, in the bipartite graph $G(\mathbb{X}, \Gamma)$. This procedure is repeated for the new sets of shared variables that come with newly introduced shared FMSO sets. Iterations stop when no new shared variables are introduced. The computational complexity of the search problem increases with the number of shared variables. However, in practice, subsystems are generally designed so that their links are quite weak, hence they share few variables. This makes the proposed approach applicable to complex dynamic systems made up of several subsystems.

The distributed architecture requires that each of the local diagnosers perform their calculations independently, so it is not necessary for each local diagnoser to share its local model since only measures are shared, which has a confidentiality advantage. Algorithm 4.1 demonstrates that it is possible to obtain the same global FMSO sets and hence analytical redundancy relations as the centralized approach while maintaining the confidentiality of each local model.

4.2.2 Distributed Generation of an Optimized Set of Global FMSO Sets

If the residuals corresponding to all the global FMSO sets were generated and used on-line to monitor the system, they would obviously achieve maximal detectability and isolability. However, not all of them are necessary and it is more efficient to minimize their number while maintaining the same property.

The aim of this section is to obtain a set of distributed local diagnosers that together make the entire system completely diagnosable through local and compound FMSO sets. These local diagnosers are designed to achieve maximal diagnosability with minimal communication between subsystems. First, local FMSO sets are determined for every subsystem Σ_i. If these are not sufficient to detect and isolate all of the faults in F_i, then a set of compound FMSO sets is determined to achieve full diagnosability for all the faults in F_i, considering constraints of distance and amount of communication between subsystems [Pérez 2016].

The diagnosers design is done off-line and consists of the steps given in Algorithm 4.2, performed for each subsystem Σ_i, $i = 1...n$. The procedure to compute 'good' compound FMSO sets starting with φ^* as a root FMSO set makes use of an optimization heuristic based on the number of shared variables. In Algorithm 4.2 [Pérez 2016], the term 'best' is hence used in the sense of this heuristic.

Algorithm 4.2: Generation of local diagnosers.

for *i=1...n* **do**
- $\Phi_i = \emptyset$;
- $\Phi_i^l \leftarrow$ Calculate local FMSO sets of Σ_i;
- **if** *there is any fault* $f \in F_i$ *not locally detectable or not locally isolable with the set of local FMSO sets* Φ_i^l **then**
  - $\Phi_i^s \leftarrow$ Calculate shared FMSO sets of Σ_i;
  - $\Psi_i^s \leftarrow$ Calculate shared CMSO sets of Σ_i;
- **while** *it exists* $f \in F_i$ *that is not detectable or isolable* **do**
  - Let $\varphi^* \in \Phi_i^s$ such that $f \in F_{\varphi^*}$ be the 'best' (not already selected) shared FMSO set of Φ_i^s ;
  - Label φ^* as root FMSO set: $\varphi_r \leftarrow \varphi^*$;
  - Let $X_{\varphi_r}^s$ be the set of shared variables of φ_r;
  - $\Phi_i^{c*} \leftarrow$ Find a 'good' compound FMSO set including φ^* by always selecting the 'best' shared FMSO sets to cover newly introduced shared variables;
  - $\Phi_i \leftarrow \Phi_i \cup \Phi_i^{c*}$;
  - $\Phi_i^{l*} \leftarrow$ Find a minimal cardinality set of local FMSO sets achieving the same diagnosability as all local FMSO sets;
  - $\Phi_i \leftarrow \Phi_i \cup \Phi_i^{l*}$;
- $ARR_i \leftarrow$ Generate the analytical redundancy relations of LD_i from the FMSO sets in Φ_i;

Algorithm 4.2 provides the principles to produce a minimal cardinality set of global FMSO sets while minimizing subsystems interactions. The optimization problem will be formalized and solved with a heuristic search algorithm in the next chapter.

4.3 Implementation of the Distributed Diagnoser Design

After the **offline** design of the local diagnosers performed with algorithm 4.2, the **online** implementation of the distributed diagnoser relies on the bank of residual generators ARR_i selected for each local diagnoser $LD_i, i = 1, \ldots, n$, fed by measured signals from their corresponding subsystems. As shown in Figure 4.1, fault isolation is carried out after fault detection using local fault signature matrices according to Definition 31. Let us notice that there are no upper hierarchical levels.

Definition 31 (Fault Signature Matrix of a subsystem)**.** *Given a set ARR_i composed of n_i^r ARRs and F_i the set of considered n_i^f faults for the subsystem Σ_i and consider the function $ARR_i \times F_{j,i} \longrightarrow 0, 1$, then the signature of a fault $f \in F_i$ is the binary vector $FS_i(f) = [\tau_1, \tau_2, ...\tau_{n_i^r}]^T$ where $\tau_k = 1$ if f is involved in the equations used to form $arr_k \in ARR_i$, otherwise $\tau_k = 0$. The signatures of all the faults in F_i together constitute the fault signature matrix FSM_i for subsystem Σ_i, i.e. $FSM_i = [FS_i(f_1), \ldots, FS_i(f_{n_i^f})]^T$.*

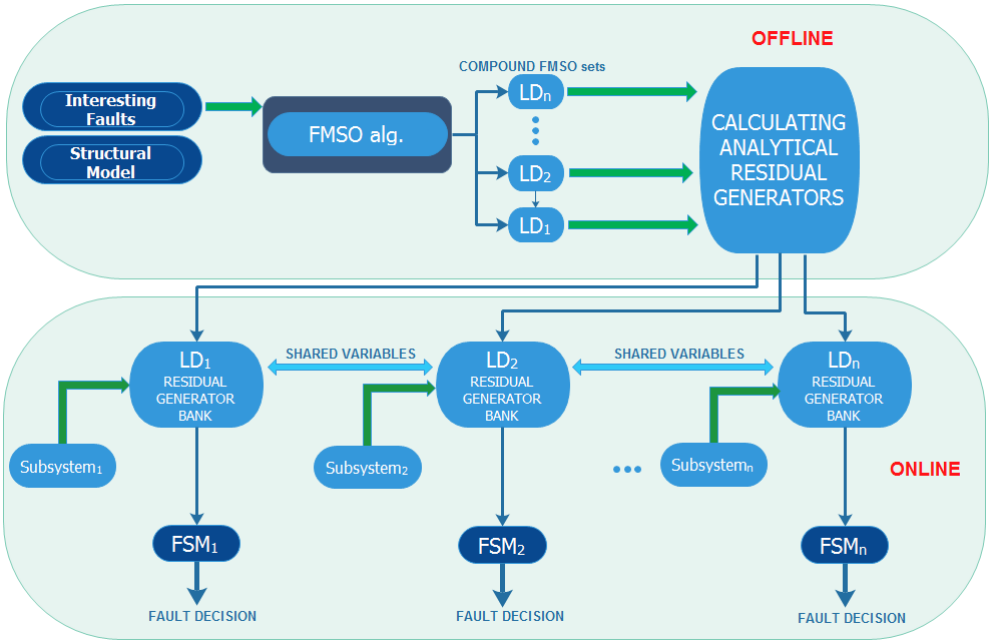


Figure 4.1: Scheme of a distributed diagnoser.

4.4 Application to the Four-Tank System

The distributed diagnoser is illustrated on the four-tank system shown in Figure 4.2.

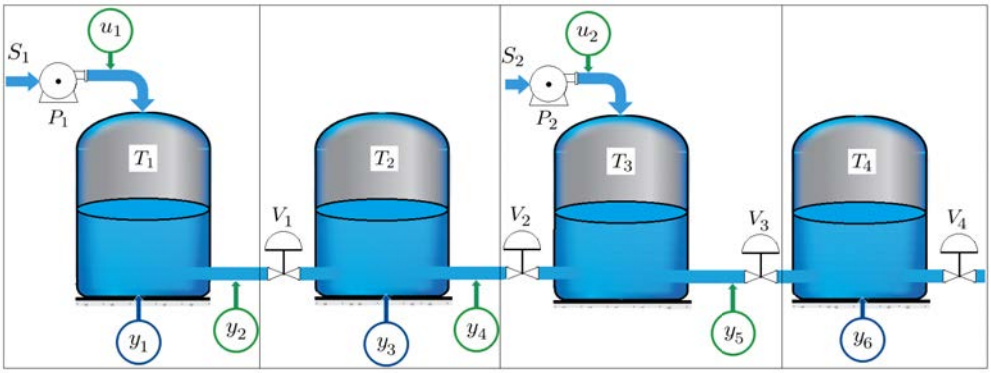


Figure 4.2: Four coupled tank benchmark from [Khorasgani 2015].

the summary of components is the following:

- tanks T_1, T_2, T_3 and T_4,
- pumps P_1 and P_2,
- proportional valves V_1, V_2, V_3 and V_4,
- sensors for variables y_1, y_2, y_3, y_4, y_5 and y_6,
- two flow sensors for the input flow rates u_1 and u_2.

The global model $\Sigma(z, x, \mathtt{f})$ for this system is composed of twenty equations, the known variables $Z = \{u_1, u_2, y_1, y_2, y_3, y_4, y_5, y_6\}$, the unknown variables $X = \{\dot{p}_1, p_1, \dot{p}_2, p_2, \dot{p}_3, p_3, \dot{p}_4, p_4, q_{in1}, q_{in2}, q_1, q_2, q_3, q_4\}$ and the set of system faults $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$. All proportional valves are assumed to be fully open.

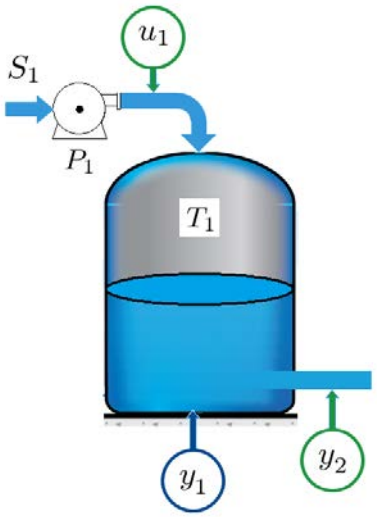
The structural model representation of the four-tank system is given in Table 4.1.

Eq	Unknown													Faults						Known							
	$\dot{p}_1$	p_1	q_{i1}	$\dot{p}_2$	$\dot{p}_3$	q_{i2}	$\dot{p}_4$	q_4	q_2	p_3	q_3	p_4	q_1	p_2	f_1	f_2	f_3	f_4	f_5	f_6	u_1	u_2	y_1	y_2	y_3	y_4	y_5	y_6
e_1	X		X										X		X													
e_2		X											X	X		X												
e_3	X	X																										
e_4			X																		X							
e_5		X																					X					
e_6													X											X				
e_7				X					X				X			X												
e_8									X	X				X				X										
e_9				X										X														
e_{10}														X														
e_{11}									X																			
e_{12}					X	X			X		X																	
e_{13}										X	X	X							X									
e_{14}					X					X																		
e_{15}						X															X							
e_{16}											X																	X
e_{17}							X	X			X									X								
e_{18}								X				X																
e_{19}							X					X																
e_{20}												X																X

Table 4.1: Structural representation of four coupled tank benchmark.

Now, we consider each tank with outlet pipe as a subsystem so this system is decomposed into four subsystems. Tanks 1 and 3 have inflows and there is a set of 6 measurements:

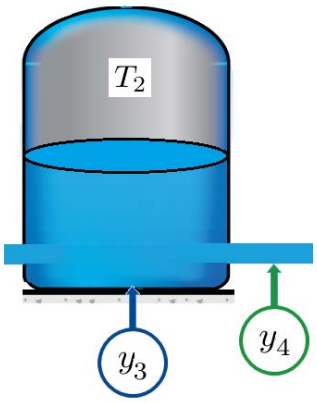
The first subsystem $\Sigma_1(z, x, \mathtt{f})$ is described by the drawing and the set of equations shown in Figure 4.3. Therefore we have $\Sigma_1 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ which is the set of equations, $X_1 = \{\dot{p_1}, p_1, p_2, q_{in1}, q_1\}$ is the set of subsystem unknown variables, $Z_1 = \{u_1, y_1, y_2\}$ is the set of subsystem measurements and $F_1 = \{f_1, f_2\}$ is the set of faults for this subsystem.



$$e_1 : \dot{p}_1 = \frac{1}{C_{T_1} + f_1}(q_{in1} - q_1)$$
$$e_2 : q_1 = \frac{p_1 - p_2}{R_{P_1} + f_2}$$
$$e_3 : p_1 = \int \dot{p}_1 dt$$
$$e_4 : q_{in1} = u_1$$
$$e_5 : p_1 = y_1$$
$$e_6 : q_1 = y_2$$

Figure 4.3: Scheme and equations of subsystem Σ_1.

Similarly, the second subsystem $\Sigma_2(z, x, \mathtt{f})$ is described by the drawing and the set of equations shown in Figure 4.4.



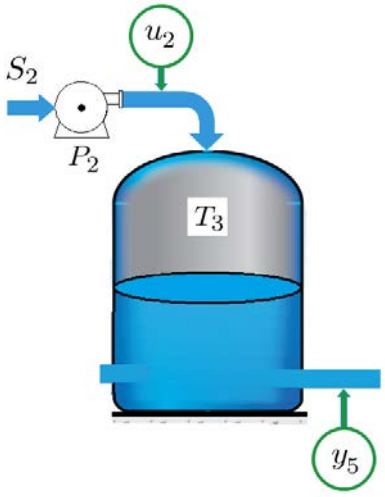
$$e_7 : \dot{p}_2 = \frac{1}{C_{T_2} + f_3}(q_1 - q_2)$$
$$e_8 : q_2 = \frac{p_2 - p_3}{R_{P_2} + f_4}$$
$$e_9 : p_2 = \int \dot{p}_2 dt$$
$$e_{10} : p_2 = y_3$$
$$e_{11} : q_2 = y_4$$

Figure 4.4: Scheme and equations of subsystem Σ_2.

This subsystem is defined by $\Sigma_2 = \{e_7, e_8, e_9, e_{10}, e_{11}\}$ that is the set equations, $X_2 = \{\dot{p_2}, p_2, p_3, q_1, q_2\}$ is the set of subsystem unknown variables, $Z_2 = \{y_3, y_4\}$ is

the set of subsystem measurements, and $F_2 = \{f_3, f_4\}$ is the set of faults for this subsystem.

The third subsystem $\Sigma_3(z, x, \mathtt{f})$ is described by the drawing and the set of equations shown in Figure 4.5. In this subsystem there is no measurement of tank level T_3.

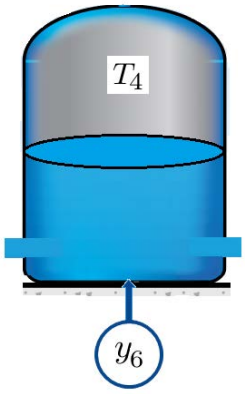


$$e_{12}{:}\dot{p}_3 = \frac{1}{C_{T_3}}(q_{in2} + q_2 - q_3)$$
$$e_{13}{:}q_3 = \frac{p_3 - p_4}{R_{P_3} + f_5}$$
$$e_{14}{:}p_3 = \int \dot{p}_3 dt$$
$$e_{15}{:}q_{in2} = u_2$$
$$e_{16}{:}q_3 = y_5$$

Figure 4.5: Scheme and equations of subsystem Σ_3.

This subsystem is defined by $\Sigma_3 = \{e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\}$ that is the set of equations, $X_3 = \{\dot{p}_3, p_3, p_4, q_{in2}, q_2, q_3\}$ is the set of subsystem unknown variables, $Z_3 = \{u_2, y_5\}$ is the set of subsystem measurements, and $F_3 = \{f_5\}$ is the set of faults for this subsystem.

Finally, the fourth subsystem $\Sigma_4(z, x, \mathtt{f})$ is described by the scheme and the set of equations shown in Figure 4.6.



$$e_{17}{:}\dot{p}_4 = \frac{1}{C_{T_4} + f_6}(q_3 - q_4)$$
$$e_{18}{:}q_4 = \frac{p_4}{R_{P_4}}$$
$$e_{19}{:}p_4 = \int \dot{p}_4 dt$$
$$e_{20}{:}p_4 = y_6$$

Figure 4.6: Scheme and equations of subsystem Σ_4.

This subsystem is defined by $\Sigma_4 = \{e_{17}, e_{18}, e_{19}, e_{20}\}$ that is the set of equations, $X_4 = \{\dot{p}_4, p_4, q_3, q_4\}$ is the set of subsystem unknown variables, $Z_4 = \{y_6\}$ is the set

of subsystem measurements, and $F_4 = \{f_6\}$ is the set of faults for this subsystem.

The distribution of equations of Σ, known variables Z, unknown variables X and faults F for the four subsystems $\Sigma_i, i = 1, 2, 3$ and 4 is summarized below in Equations (4.1), (4.2), (4.3) and (4.4).

$$\Sigma_1 = \begin{cases} \Sigma_1 = \{e_1, e_2, e_3, e_4, e_5, e_6\} & F_1 = \{f_1, f_2\} \\ X_1 = \{\dot{p_1}, p_1, p_2, q_{in1}, q_1\} & Z_1 = \{u_1, y_1, y_2\} \end{cases} \tag{4.1}$$

$$\Sigma_2 = \begin{cases} \Sigma_2 = \{e_7, e_8, e_9, e_{10}, e_{11}\} & F_2 = \{f_3, f_4\} \\ X_2 = \{\dot{p_2}, p_2, p_3, q_1, q_2\} & Z_2 = \{y_3, y_4\} \end{cases} \tag{4.2}$$

$$\Sigma_3 = \begin{cases} \Sigma_3 = \{e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\} & F_3 = \{f_5\} \\ X_3 = \{\dot{p_3}, p_3, p_4, q_{in2}, q_2, q_3\} & Z_3 = \{u_2, y_5\} \end{cases} \tag{4.3}$$

$$\Sigma_4 = \begin{cases} \Sigma_4 = \{e_{17}, e_{18}, e_{19}, e_{20}\} & F_4 = \{f_6\} \\ X_4 = \{\dot{p_4}, p_4, q_3, q_4\} & Z_4 = \{y_6\} \end{cases} \tag{4.4}$$

According to the operational procedure of Section 4.2.1, with Algorithm 4.1 it is possible to get the set of all global FMSO sets Φ from the set of local FMSO sets Φ^l, shared FMSO sets Φ^s and shared CMSO sets Ψ^s.

Running the Algorithm 4.1, first we compute local FMSO sets Φ_i^l, shared FMSO sets Φ_i^s and shared CMSO sets Ψ_i^s of each subsystem $\Sigma_i, i = 1, \ldots, 4$ as shown in Table 4.2. Then with each shared FMSO set as root FMSO set, we find all compound FMSO sets $\varphi \in \Phi^c$ for the four-tank system as if a global model was available.

As illustration, in the subsystem Σ_1, considering the shared FMSO set φ_1 as a root FMSO set with the set of $X_{\varphi_1}^s = \{q_1, p_2\}$, a compound FMSO set is computed iteratively as the set $\varphi^c = \varphi_1 \cup \varphi_5 \cup \varphi_6 \cup \psi_3 \cup \varphi_7 \cup \psi_4 \cup \psi_6$, with $X_{\varphi^c}^s = \{q_1, p_2, q_2, p_3, q_3, p_4\}$. Each shared variable x^s is covered by two shared FMSO/CMSO sets as it is shown in the corresponding subgraph of Figure 4.7. As a result, the compound FMSO set φ' obtained is $\{e_2, e_5, e_7, e_8, e_9, e_{11}, e_{13}, e_{16}, e_{20}\}$. Considering all the possible root FMSO sets, 164 compound FMSO sets are computed for this system. Added to $\varphi_4 = \{e_1, e_3, e_4, e_5, e_6\} \in \Phi_1^l$, which is a local FMSO set for subsystem Σ_1, the 165 global FMSO sets are found for Φ.

4.4.1 Distributed Diagnosis

Given a set of faults, measurements and local models for every subsystem, we now construct local diagnosers that together make the entire system completely diagnosable. Using the Algorithm 4.2 and definitions of Chapter 2, we can develop a local full diagnosis for every subsystem. Computing the set of local FMSO sets Φ_i^l, $i = 1, 2, 3$ and 4 and adding subsets of shared variables to find the set of shared FMSO sets Φ_i^s for each subsystem Σ_i, $i = 1, 2, 3$ and 4 in Equations (4.1), (4.2), (4.3) and (4.4), we find FMSO sets whose fault support cover all faults as it is

Table 4.2: Local FMSO sets Φ_i^l, shared FMSO sets Φ_i^s and shared CMSO sets: Ψ_i^s, $i = 1, \ldots, 4$.

$\Phi_1^s = \{\varphi_1, \varphi_2, \varphi_3\}$, $\Phi_1^l = \{\varphi_4\}$, $\Psi_1^s = \{\psi_1\}$,
$\Phi_2^s = \{\varphi_5, \varphi_6\}$, $\Phi_2^l = \emptyset$, $\Psi_2^s = \{\psi_2, \psi_3\}$,
$\Phi_3^s = \{\varphi_7\}$, $\Phi_3^l = \emptyset$, $\Psi_3^s = \{\psi_4, \psi_5\}$,
$\Phi_4^s = \{\varphi_8\}$, $\Phi_4^l = \emptyset$, $\Psi_4^s = \{\psi_6\}$,

Φ_i	X^s						F_i
Σ_1	q_1	p_2	q_2	p_3	q_3	p_4	F_1
$\varphi_1 = \{e_2, e_5\}$	X	X					$\{f_2\}$
$\varphi_2 = \{e_1, e_3, e_4, e_5\}$	X						$\{f_1\}$
$\varphi_3 = \{e_1, e_2, e_3, e_4\}$	X	X					$\{f_1, f_2\}$
$\varphi_4 = \{e_1, e_3, ..., e_6\}$	X	X					$\{f_1\}$
$\psi_1 = \{e_6\}$	X						

Σ_2	q_1	p_2	q_2	p_3	q_3	p_4	F_2
$\varphi_5 = \{e_8\}$		X	X	X			$\{f_4\}$
$\varphi_6 = \{e_7, e_9\}$	X	X	X				$\{f_3\}$
$\psi_2 = \{e_{10}\}$		X					
$\psi_3 = \{e_{11}\}$			X				

Σ_3	q_1	p_2	q_2	p_3	q_3	p_4	F_3
$\varphi_7 = \{e_{13}\}$				X	X	X	$\{f_5\}$
$\psi_4 = \{e_{16}\}$					X		
$\psi_5 = \{e_{12}, e_{14}, e_{15}\}$			X	X	X		

Σ_4	q_1	p_2	q_2	p_3	q_3	p_4	F_4
$\varphi_8 = \{e_{17}, e_{18}, e_{19}\}$					X	X	$\{f_6\}$
$\psi_6 = \{e_{20}\}$						X	

shown in Table 4.3.

These results demonstrate that all considered faults can be detected and isolated. For example in Σ_1, detectability is achieved for f_1 using $\varphi_4 \in \Phi_i^l$ of Table 4.2 (no additional measurement is needed). For f_2, detectability is achieved obtaining a compound FMSO set $\varphi_9 \in \Phi_1^c$ lumping $\varphi_1 \in \Phi_1^s$ (as root FMSO set) with $\psi_1 \in \psi_1^s$ and $\psi_2 \in \Psi_2^s$.

Figure 4.8 shows a scheme of the proposed distributed diagnosis architecture for this system: the four subsystems with their physical interactions are represented on the left. On the right, each local diagnoser LD_i is represented as a rectangle with selected FMSO sets. The arrows from the corresponding subsystems symbolize the direct measurement of local variables by the LD_i's, while the arrows between

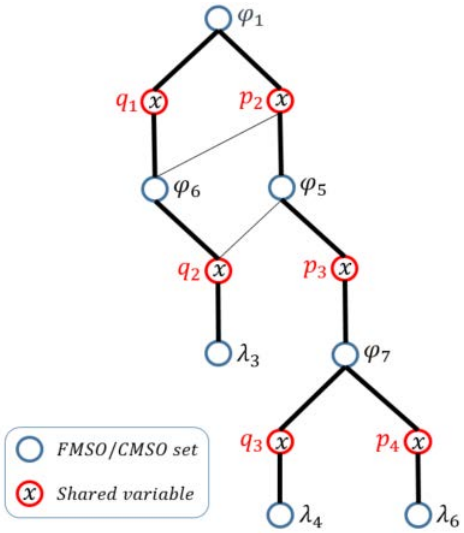


Figure 4.7: Subgraph of φ'.

Table 4.3: Optimal compound FMSO sets Φ_i^c, $(i = 1..4)$ for distributed diagnosis.

$\Phi_1^c = \{\varphi_9\}$	F_{Φ_1}
$\varphi_9 = \{e_2, e_5, e_6, e_{10}\}$	$F_{\varphi_9} = \{f_2\}$
$\Phi_2^c = \{\varphi_{10}, \varphi_{11}\}$	F_{Φ_2}
$\varphi_{10} = \{e_6, e_7, e_9, e_{10}, e_{11}\}$	$F_{\varphi_{10}} = \{f_3\}$
$\varphi_{11} = \{e_8, e_{10}, e_{11}, e_{13}, e_{16}, e_{20}\}$	$F_{\varphi_{11}} = \{f_4\}$
$\Phi_3^c = \{\varphi_{12}\}$	F_{Φ_3}
$\varphi_{12} = \{e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{20}\}$	$F_{\varphi_{12}} = \{f_5\}$
$\Phi_4^c = \{\varphi_{13}\}$	F_{Φ_4}
$\varphi_{13} = \{e_{16}, e_{17}, e_{18}, e_{19}, e_{20}\}$	$F_{\varphi_{13}} = \{f_6\}$

the local diagnosers account for shared information necessary to complete local diagnosis.

4.5 Conclusion

In this chapter, a distributed fault diagnosis method is presented. Distributed diagnosis is of interest for on-board systems as a way to reduce computational costs or for large geographically distributed systems that require to minimize data transfer. The FMSO set concept is central to this approach. An FMSO set can be directly used to construct one ARR or residual generator, as compared to MTES that lead to several. We believe that FMSO sets represent a more practical solution in distributed contexts in which communication must be minimized. This chapter

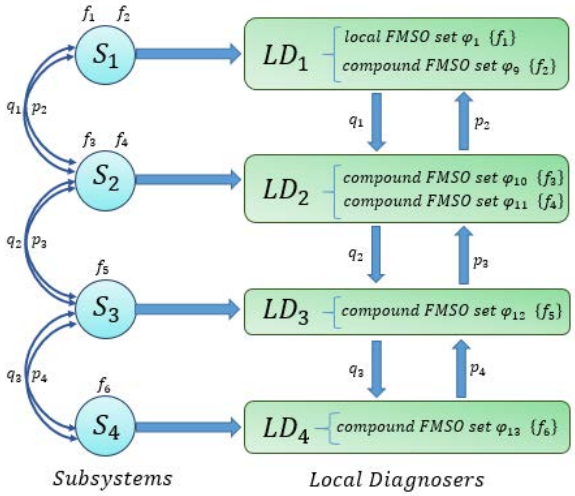


Figure 4.8: Scheme of the decentralized diagnosis designed.

then provides the results that show that all the global FMSO sets, i.e. those that would be obtained along a centralized approach, can be obtained from computations performed at the level of local subsystems plus a search procedure. This is possible thanks to the concept of local FMSO set and shared FMSO/CMSO sets. The operational procedures for deriving in a distributed way all the global FMSO sets and a 'good' set of global FMSO sets are presented. These are illustrated with the four-tank benchmark.

The next chapter presents the optimization problem of generating a minimal cardinality set of compound FMSO sets that minimize subsystem interactions. The aim of the next chapter is hence to obtain optimal local diagnosers that guarantee the same properties as a global diagnoser.

Part III

Optimization Algorithms for Decentralized and Distributed Fault Diagnosis

Chapter 5

A* Algorithms for Optimized Distributed Structural Diagnosis

Contents

5.1 Introduction

The main objective of this chapter is to proposes a set of optimization methods for selecting diagnosis tests generated from FMSO sets. These methods optimize the diagnosis process since all the tests do not need to be built but only those necessary to achieve detectability and isolability of the interesting faults.

The *optimal test selection problem* as known in the literature aims at minimizing the cost of the tests while satisfying some isolability constraints. This problem is

also related to the well-known *sensor placement problem*, which searches the minimum cost sensor configuration that satisfies a given set of fault diagnosis specifications. Here, this problem is managed considering that each local diagnoser has a set of candidate sensors (that may be common to other local diagnosers through shared variables) that can be optimally selected to complete its diagnostic requirements.

We propose three algorithms to solve the test selection problem. The common feature of the three algorithms is that they are cast in a heuristic search framework.

The first algorithm proposes a solution for the distributed architecture that can however also used for a classical centralized architecture. Let us remember that the distributed approach presented in Chapter 4, produces local FMSO sets for the subsystems and, as shown in Chapter 2, local FMSO sets are also global FMSO sets. So, our selection algorithm starts from a set of global FMSO sets that may be generated following the distributed method of Algorithm 4.1 given in subsection 4.2.1 of chapter 4 or in a centralized way from the whole model of the system. In the latter case, which is the one considered in this thesis, the set of FMSO sets given as input is already reduced thanks to the isolation on request strategy.

The second and third algorithms are designed for a distributed architecture. They start from shared FMSO sets obtained for each subsystem and generate only those global FMSO sets that are required to reach detectability and isolability for each fault in each subsystem. They differ in the way every subsystem is processed: in parallel for the second and iteratively for the third.

The problem of optimal selection of shared FMSO and CMSO sets can also be formulated as a combinatorial optimization problem. Such formulation has already been proposed using binary integer programming [Bagajewicz 2004, Sarrate 2007, Rosich 2009b] but not in a distributed context.

In this thesis, we claim that it is possible and interesting to analyze optimal test selection problem as a planning and more specifically as a pathfinding problem starting from an initial node whose state is totally ambiguous diagnosis to a goal node achieving maximum diagnosability. This point of view is one contribution of this thesis.

Planning consists in organizing optimally a limited set of actions in order to reach one goal. Actions generally consume and produce resources, have a cost, and the goal is expressed as a desired value for some of these resources. From a diagnosis perspective, planning can also be considered as driving a process modeled by an automaton toward a goal state, in an optimal manner, when all transitions are controllable. Each state then represents a tuple of values, one per resource, and transitions derive from the possible actions. In these terms, the problem consists in finding a shortest path in a possibly huge weighted oriented graph, from an initial node to a set of possible final nodes.

In this chapter we propose to use A*-like algorithms to solve the different variants of the test selection problem and optimally select the best FMSO sets for each sub-system to achieve the best possible fault detectability and isolability.

Despite the NP-hardness of the problem, efficient algorithms can be proposed, as variants of the A*. The A* algorithm is nothing but a depth-first search, guided to

the goal by some heuristic function, i.e. a lower-bound on the distance to the goal, available at each node. In practice, provided heuristics are smartly designed, this approach performs much better than the worst case bound, that requires exploring the whole graph.

Some hypotheses and properties must be recalled: first, we assume that there is no exchange of diagnosis information among the local diagnosers, only exchange of measurements; secondly, as demonstrated in previous chapters, FMSO sets are needed to elaborate diagnosis tests which take the form of ARRs in this thesis, so test selection is the same as ARR selection and exactly corresponds to FMSO set selection. This is why in the following, *FMSO sets* are sometimes called *tests* and we may also refer to *ARRs*.

Two approaches are investigated:

- In the decentralized case, an optimization approach chooses among the global FMSO sets generated through the decentralized hierarchy to ensure detectability and isolability for all the faults of the system. The selected global FMSO sets are assigned to their corresponding subsystem in the hierarchy to form the decentralized diagnostic system.

- In the distributed case, only shared FMSO are available for the computations. Two algorithms are proposed. Algorithm *FirstLocalThenComplete* uses an A* strategy in parallel for each subsystem and then considers the selected shared FMSO sets as root FMSO sets to form compound FMSO sets that ultimately are global FMSO sets of the system. We refer to this operation as to *complete* a shared FMSO set. Algorithm *IterativeFindAndComplete* applies an A* strategy to one subsystem and completes the set of selected shared FMSO sets with the shared FMSO/CMSO sets of the other subsystems. Then the algorithm processes another subsystem (if not already diagnosable from the previous completion operations), and so on until all the subsystems have been processed. As opposed to the first optimization algorithm, the two algorithms for the distributed case are themselves distributed.

5.2 Related Works for Optimal Test Selection

In this section, we propose a review of the existing works dealing with optimal test selection for fault diagnosis and then we summarize some related work with respect to planning and specifically to the A* algorithm.

5.2.1 Optimal Test Selection for Fault Diagnosis

In the litterature, optimal test selection is often associated to the problem of test prioritization that corresponds to choosing the next best test or measurement to disambiguate a faulty situation. In practice, this is an integral part of the troubleshooting task. This domain has received a lot of attention [Pattipati 1988, Pattipati 1992, Dick 1993] but as far as we know the problem is

never addressed in a decentralized/distributed framework. This is one of the main contributions of this chapter.

Solutions have been proposed through heuristic optimization techniques [Li 2007, Raghavan 1999], which are in line with the optimization approach of this thesis. The problem is formulated in a centralized context, hence the authors focus on diagnosis ambiguity reduction as the main optimization criterion, which leads them to adopt heuristics based on information theory [Kleer 1987, **?**]. [Faure 1999, Olive 2003, Travé-massuyès 2013] proposed solutions based on a dictionary of fault signatures supporting heuristic optimization techniques or the computation of the tests entropy. In [Ressencourt 2006], hybrid system simulation techniques, based on the *Modelica* language, are used to build the dictionary of fault signatures from faulty models. The methods known as *Diagnostic Test Prioritization Techniques* are based on the Information Gain heuristic. They maximize the diagnostic information gain per test and increase the rate at which diagnosis quality improves [Gonzalez-Sanchez 2011b] but they are often limited by their complexity. Nevertheless, we can mention the gReedy diAgnostic Prioritization by ambiguiTy Reduction (RAPTOR) method [Gonzalez-Sanchez 2011a] as a instance that achieves to restrain this issue. It relies on a quite a intuitive diagnosis ambiguity heuristic and diagnostic performance is expressed in terms of a cost metric that measures the number of inspected components that are not the faulty one, i.e. the wasted effort.

Unlike the above mentioned works, the optimization problems that we formulate in this chapter target decentralized and distributed diagnosis architectures and this is why they are driven by the requirement of minimizing the cost of communication between subsystems, although diagnosis ambiguity reduction is obviously also present in the optimality criterion.

As said before, optimal test selection is very close to sensor selection and this is why it is also interesting to review the litterature. Let us note that none of the reviewed works cast the problem in a decentralized/diatributed diagnosis framework.

In [Travé-Massuyès 2006] a method for characterizing and determining the minimal additional sensors that guarantee a specified degree of diagnosability is proposed. The diagnosability degree is achieved by removing iteratively sensors assuming that initially all possible sensors are available. The paper also characterizes Minimal Additional Sensor Sets (MASS), which guarantee maximal discrimination level. The work calls for sophisticated optimization methods to solve the MASS optimization problem in its general form. The optimal sensor selection is also considered in [Rosich 2007] starting with no sensors and iteratively adding sensors while the MSO sets are incrementally generated. The main improvement is the uselessness of generating all the MSO sets.

[Commault 2008] uses the notion of input separator in the structural model represented by a graph. The paper presents a set of conditions about the number of additional sensors measuring variables between fault nodes and input separators in the associated graph. This work is limited to linear structured systems and presents computational complexity issues for large models, also, there are some

strong assumptions that are difficult to fulfill in practical situations relating to residuals that must be sensitive to a single fault and must be measurable.

[Krysander 2008b] presents an algorithm for sensor placement for detectability and isolability that takes into account an isolability specification and the possible sensor locations. This algorithm uses a structural model of the system and is based on the partial order defined by the well-constrained part of the Dulmage-Mendelsohn decomposition to compute all minimal sensor sets that make, as far as possible and desired, faults isolable from each others. [Yassine 2008] uses an alternative structural model decomposition, based on gathering equations that can not be isolable.

In [Sarrate 2012] the sensor placement problem applied to distribution networks is also addressed using the structural analysis framework. A definition of the isolability index as a measurement of the fault diagnosis performance achievable in a given system is presented for setting up a sensor placement problem based on a fault diagnosis performance maximization criterium.

Finally, [Leal 2015] presents an approach for diagnosability analysis and sensor placement based on genetic algorithms. The approach selects the minimum number of MTES to be selected for generating diagnosis tests. Genetic algorithms appear as an efficient tool to solve the combinatorial problem for the selection of MTES given a structural model. This goal of this work is quite related to ours but it is formulated in a centralized framework. In our work, we prefer to use the concept of FMSO set which is better suited to decentralized and distributes diagnosis architectures because unlike MTES, every FMSO set point to one single test.

5.2.2 Search Algorithms

To solve the pathfinding problem, search algorithms are used. The most general search algorithms are based on brute-force search since they do not require any domain specific knowledge. As examples we can cite Breadth-First search, Uniform-Cost search, Depth-First Search [Korf 2010].

Search problems occur very frequently in graph theory. We recall here some basic notions of graph theory. The goal is to give some clues in order to understand the main idea of search algorithms but more details can be found in [Bondy 1976].

A graph G is an ordered triple $(V(G), E(G), \Gamma)$ consisting in a nonempty set $V(G)$ of vertices or nodes, a set $E(G)$, disjoint from $V(G)$ of edges and an incidence function Γ that associates with each edge of G an unordered pair of vertices. A search problem is defined by a set of states (often represented by nodes in a graph); a start state (or root node) and a set of goal states. A successor function provides a mapping from a state to a set of successor states. A path in G is a finite non-null sequence $P = n_0 e_1 n_1 e_2 n_2 \dots e_k n_k$, whose terms are alternately vertices and edges, such that, for $i \leq i \leq k$, the ends of e_i are n_{i-1} and n_i and all vertices are distinct. The shortest path problem consists in associating a real number $w(e)$ with each edge e of G, called its weight (also named cost or score). Then G, together with these weights on its edges, is called a weighted graph. The goal is to find a path of

minimum weight connecting two specified vertices n_0 (the root node) and n_f (any node in the set of goal states).

In the brute-force algorithms, at each step, the next node n to be expanded is the one whose cost $g(n)$ is the lowest, where $g(n)$ is the sum of the edge costs from the root node n_0 to node n. In order to solve larger problems, domain-specific knowledge must be added to improve search efficiency [Atallah 2009].

In AI, heuristic search has a general meaning and a more specialized technical meaning. In the general sense, the term heuristic is used for any advice that is often effective, but is not guaranteed to work in every case. Here, the term heuristic refers to the special case of a heuristic evaluation function. In a single-agent path-finding problem, a heuristic evaluation function estimates the cost of an optimal path between a pair of states. For a fixed goal state, a heuristic evaluation $h(n)$ is a function of a node n, which estimates the distance from node n to the goal state. For example, Euclidean or airline distance is an estimate of the highway distance between a pair of locations [Korf 2010].

Most heuristic functions are derived by generating a simplified version of the problem to be solved, then using the cost of an optimal solution to the simplified problem as a heuristic evaluation function for the original problem. The simplest of these algorithms is a pure heuristic search and expands nodes in order of their heuristic values $h(n)$ [Doran 1966].

The A* algorithm, initially presented in [Hart 1968], combines features of uniform-cost search and pure heuristic search to efficiently compute optimal solutions. The next section will present it more precisely. The goal here is to discuss how the A* algorithm can been adapted for distributed problems. Pathfinding, or planning a route to a destination that avoids obstacles, is a classic problem in AI. However, in the case of distributed systems, pathfinding problems for a team of agents or for different subsystems are usually solved in two ways [Standley 2012].

In global search approaches, the entire set of agents is viewed as a single entity and paths are found for all agents simultaneously. Alternatively, in decoupled approaches, paths are found for each agent one at a time, and information about the paths of other agents is used to ensure that no paths conflict. Global search approaches typically have the advantage of being complete, meaning that they always eventually find a solution to any problem if a solution exists, but they are often intractable for even small numbers of agents (about 20). On the other hand, decoupled approaches are fast, but incomplete.

[Hearn 2005] presents an algorithm that chooses a fixed ordering of agents, and plans a path for each agent in turn that avoids conflicts with previously computed paths by checking against a reservation table. Unfortunately, some agents never reach their destinations because the paths found for previous agents in the fixed order can make finding paths for subsequent agents impossible.

[Silver 2005] presents an algorithm called LRA. In LRA, a path is computed for each agent independently, and conflicts are not discovered until execution. If the algorithm discovers that following a solution one step further results in a conflict, the algorithm re-plans the path of one of the conflicting agents.

5.3 The A^* Algorithm

5.3.1 Basic Notions

A* presented in [Hart 1968] as "A star" is a computer algorithm that is widely used in pathfinding and graph traversal, the process of plotting an efficiently traversable path between points, called nodes. As already said, the A* algorithm combines features of uniform-cost search and pure heuristic search to effectively compute optimal solutions. Noted for its performance and accurancy, A* enjoys widespread use. This algorithm is an extension of Dijkstra's Algorithm achieving better performance (with respect to time) by using heuristics. As A* traverses the graph, it follows a path of the lowest known cost, keeping a sorted priority queue of alternate path segments along the way. Similar to greedy best-first search, it is more accurate because A* takes into account the nodes that have already been traversed. If, at any point, a segment of the path being traversed has a higher cost than another encountered path segment, it abandons the higher-cost path segment and traverses the lower-cost path segment instead. This process continues until the goal is reached [Zeng 2009].

The objective of the algorithm is to expand the smallest possible number of nodes in the search for an optimal path. For this, an informed decision must be taken about which node to expand next, in order to avoid expanding nodes that are not in an optimal path. On the other hand, if the algorithm ignores nodes that could be on an optimal path, sometimes it will not be able to find such a path and, therefore, not be admissible.

More precisely, A* is a best-first search in which the score [1] associated with a node is $f(n) = g(n) + h(n)$, where:

- $g(n)$ is the score of the path from the initial state n_0 to the current node n,
- $h(n)$ is the heuristic estimate of the score of a path from node n to a goal. $h(n)$ is used to approximate the distance from the current location to the goal state. This function is distinct because it is a mere estimation rather than an exact value. The more accurate the heuristic the faster the goal state is reached and the higher accuracy.
- $f(n) = g(n) + h(n)$ is the current approximation of the shortest path to the goal. f is called the *evaluation function.* $f(n)$ is calculated for any node n to determine which node should be expanded next.

At each point a node with lowest score value is chosen for expansion. Ties among nodes of equal score value are broken in favor of nodes with lower score values. The algorithm terminates when a goal node is chosen for expansion. The main drawback of A*, and indeed of any best-first search, is its memory requirement [Korf 2010].

Typical implementations of A* use a priority queue to perform the repeated selection of minimum (estimated) score nodes to expand. This priority queue is

[1] In the following, we use the term *score* but the term *cost* is also widely used in the literature.

known as the OPEN list. At each step of the algorithm, the node with the lowest $f(n)$ value is removed from the queue, the f and g scores of its neighbors are updated accordingly, and these neighbors are added to the queue. The algorithm continues until a goal node has a lower f score than any node in the queue, or until the queue is empty. The f score of the goal is then the length of the shortest path, since h at the goal is zero in an admissible heuristic.

The algorithm described so far gives us only the length of the shortest path. To find the actual sequence of steps, the algorithm can be easily revised so that each node on the path keeps track of its predecessor. After A* is run, the ending node will point to its predecessor, and so on, until some node's predecessor is the start node.

5.3.2 A* Algorithm Pseudo-code and Heuristic

The fundamental steps of the algorithm A* are shown below, as in the proposed implementation of [Nilsson 1998]:

1. Create a search graph G, consisting solely of the start node, n_0. Put n_0 on a list called *OPEN*.

2. Create a list called *CLOSED* that is initially empty.

3. If *OPEN* is empty, exit with failure.

4. Select the first node on *OPEN*, remove it from *OPEN*, and put it on *CLOSED*. Called this node n.

5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to n_0 in G (the pointers define a search tree and are established in Step 7).

6. Expand node n, generating the set M of its successors that are not already ancestors of n in G. Install these members of M as successors of n in G.

7. Establish a pointer to n from each of those members of M that were not already in G (i.e., not already on either *OPEN* or *CLOSED*). Add these members of M to *OPEN*. For each member, m, of M that was already on *OPEN* or *CLOSED*, redirect its pointer to n if the best path to m found so far is through n. For each member of M already on *CLOSED*, redirect the pointers of each of its descendants in G so that they point backward along the best paths found so far to these descendants.

8. Reorder the list *OPEN* in order of increasing f scores (ties among minimal f values are resolved in favor of the deepest node in the search tree).

9. Go to Step 3.

In step 7, if it is the case, pointers from a node are redirected if the search process discovers a path to that node having lower cost than indicated by the existing pointers.

5.3.3 Implementation Prerequisites for A* Based FMSO Selection

In this work, we develop a variation of the A* algorithm in a way that allows us to properly select the FMSO (and CMSO) sets [2] of for each subsystem starting from a complete set of faults until determining the best possible isolation. Thus, we have to take into account that an optimization problem is the problem of finding the *best* solution from all *feasible* solutions. We thus have to consider and define the following notions:

- **Variables to choose**: they can be continuous or discrete. For us the variables are boolean, the problem is known as a combinatorial optimization problem. Instead of choosing actions we choose which FMSO (CMSO) sets have to be included to generate our tests.

  Definition 32 (Node)**.** *A node of the graph is given by an FMSO (or CMSO) set φ and its fault support. Let us notice that the graph is not explicit.*

- **Criterion**: we have to define what *best* means. For us, the goal is to involve in each test the least number of connections between neighboring subsystems. It is possible to weight the order of each involved subsystem [Khorasgani 2015]. We will define for each algorithm the different score functions and heuristic f, g and h.

- **Constraints**: what do we call *feasible* solutions? In an A*, the feasible solutions are defined by the goal states meeting the constraints. In our problem, the goal states are defined as the states for which all the faults are isolable (and detectable). FMSO (CMSO) sets must be chosen so that the union of their test supports include all faults and that all faults are isolable one from the other.

5.3.4 A* for Decentralized/Distributed Structural diagnosis

In fault diagnosis, the first task is to detect the occurrence of a fault in the system (*fault detection*) and then to identify which of the faults has occurred (*fault isolation*). We recall here some basic definitions of these two concepts from the point of view of structural analysis.

Definition 33 (Detectable fault)**.** *A fault $f \in F$ is detectable in the system $\Sigma(z, x, \mathfrak{f})$ if there is an FMSO set $\varphi \in \Phi$ such that $f \in F_\varphi$, where the set of fault of the system is denoted by F.*

[2] Remember that we select among already global FMSO sets (the FMSO sets local to every subsystem in the hierarchy) in the decentralized case and among shared FMSO and CMSO sets in the distributed case.

The concept of isolation is based on determining the set of faults that can be isolated from a given fault.

Definition 34 (Isolable fault)**.** *Given two detectable faults f_j and f_k of F_i, $j \neq k$, f_j is isolable from f_k if there exists an FMSO set $\varphi \in \Phi$ such that $f_j \in F_\varphi$ and $f_k \notin F_\varphi$.*

Now, let us define the signature matrix $\mathcal{S}$. Tests, i.e. global FMSO sets, are associated to rows, and faults are associated to columns, including the no fault case. Entries $s_{i,j}$ of S take binary values: $s_{i,j} = 1$ if fault j is in the support of test i and $s_{i,j} = 0$ otherwise. When $s_{i,j} = 1$, we say that fault j is covered by test i. Isolability of all faults is achieved when the columns of S are all different. This is the property that the selection algorithm must achieve.

In the distributed case, let us notice that the fault support of a compound FMSO set is just the union of the shared FMSO/CMSO sets that compose it. Hence the isolability property can equivalently be checked with a matrix similar to S but that associates the rows to shared FMSO/CMSO sets, given that these are the items selected by the algorithm in this case. With slight abuse, we also refer to this matrix as the signature matrix and denote it S as well.

At the beginning of the optimization algorithm, the matrix $\mathcal{S}$ is denoted S_0 ; it is given the dimension $n_s \times n_f$ and it is empty (or filled with $0s$) and there is only one ambiguity set [3] composed of all the faults. n_s is the total number of FMSO (CMSO) sets and n_f is the number of faults. The set of ambiguity sets at step i is denoted $\mathcal{A}_i$.

At each step i, the matrix S_i has one more row filled with a new item . The chosen item covers some faults and it should improve isolability at the best, then we should have $Card(\mathcal{A}_i) > Card(\mathcal{A}_{i-1})$. The goal is therefore to choose the test that maximizes $Card(\mathcal{A}_i) - Card(\mathcal{A}_{i-1})$. This is obtained when every ambiguity set of $\mathcal{A}_{i-1}$ is partitioned into two isolable sets. We refer to this operation as the *dichotomic cut.*

Definition 35 (Isolability degree)**.** *The isolability degree at step i is defined as the cardinal of the set $\mathcal{A}_i$. The isolability degree is denoted $\mathcal{I}_{\mathcal{A}_i}$.*

Property 4. *The isolability degree is a function of the set of FMSO (CMSO) sets selected to form the matrix S.*

$$\mathcal{I}_S : \Phi^s \to \mathbb{R}^{+*}$$

Proof. At the beginning of the algorithm, the set of selected FMSO (CMSO) sets is empty. The initial state is given by $\mathcal{A}_0 = \{\{f_0, f_1, \ldots, f_n\}\}$ includes a unique ambiguity set that includes all the faults then $\mathcal{I}_S = 1$. At the end, when all the necesary FMSO (CMSO) are selected, the final state is $\mathcal{A}_f = \{\{f_0\}, \{f_1\}, \ldots, \{f_n\}\}$, then $\mathcal{I}_{\mathcal{S}_f} = n + 1$, when all the faults are isolable. In this case, $\mathcal{I}_{\mathcal{S}_i}$ can take all

[3] Ambiguity sets are composed of the faults that are not isolable.

the values between 1 and $n+1$. Otherwise, the optimization algorithm stops when maximal isolability is achieved. □

Concerning the state definition, the problem consists in building the signature matrix so that all faults (or a maximum number of faults) become optimally isolable. Along the algorithm, the state at step i is hence defined by the set of ambiguity sets $\mathcal{A}_i$. As already said, at the beginning of the algorithm, the state $\mathcal{A}_0 = \{\{f_0, f_1, \ldots, f_n\}\}$ includes a unique ambiguity set that includes all the faults.

FMSO (CMSO) sets that do not increase the isolability degree are not useful. It means that they do not partition at least one ambiguity set.

If all the faults are isolable, goal states are defined as all the states where $\mathcal{S}$ includes as many ambiguity sets as the number of faults. It then ensures that all faults are isolable. Equivalently it means that for goal states the isolability degree is equal to $n+1$.

Property 5. *If all the fauts are isolable, the state s_f in a goal state is $s_f = \{\{f_0\}, \{f_1\}, \ldots, \{f_n\}\}$.*

If all the faults are not isolable, goal states are defined as states where no additional FMSO (CMSO) set increases the isolability degree.

5.4 Decentralized Case: Global FMSO Sets Selection

In this section, we present a centralized A* algorithm that selects a minimal set of global FMSO sets obtained by applying the decentralized diagnoser design approach. Once the selection has been performed, every selected FMSO set is assigned to its original subsystem in the decentralized hierarchy.

5.4.1 Principles

The principles of our A* algorithm are the following.

- The start node n_0 has state $\mathcal{A}_0 = \mathcal{F} = \{\{f_0, f_1, \ldots, f_n\}\}$ (whole set of faults).
- A node n_i of the search graph is identified by the ambiguity set $\mathcal{A}_i$ resulting from the FMSO sets that have been used on the path from the start node n_0 to node n_i.
- The neighbors of a node are all the nodes that can be reached by selecting one FMSO set that increases the cardinal of the ambiguity set $\mathcal{A}_i$.
- A goal node has state $\mathcal{A}_f = \{\{f_0\}, \{f_1\}, \ldots, \{f_n\}\}$.

As explained in the previous sections, A* needs to determine which of its partial paths to expand into one or more longer paths. For our case, it need to determine which of the FMSO sets among the set of neighbors allow to reach the goal node with optimal score. The algorithm does so based on the score estimate $f(n_i) =$

$g(n_i) + h(n_i)$ for each node. Specifically, A* applied to global FMSO sets selects the path that minimizes:

The g score of any additional FMSO set is 1 (because we want to minimize the number of FMSO sets). Let n_i be the current node, then $g(n_i)$ is the number of global FMSO sets included in the solution at step i. It is the number of tests in the matrix $\mathcal{S}_i$ at current step.

Given a node n_i of the search graph, the heuristic value from n_i to a goal node is calculated by the following formula:

$$h(n_i) = Max_j \left\lceil \frac{ln(|\mathcal{A}_i^j|)}{ln(2)} \right\rceil, \tag{5.1}$$

where the $\mathcal{A}_i^j$ are the different ambiguity sets of the set $\mathcal{A}_i$ at step i and $|.|$ is the cardinal of the set.

This heuristic calculates the minimum number of FMSO sets that are necessary to disambiguate all the sets A_i^j of the ambiguity set $\mathcal{A}_i$. For one of these sets A_i^j, the minimal number of FMSO sets is $\left\lceil \frac{ln(|A_i^j|)}{ln(2)} \right\rceil$. Hence for all the sets of $\mathcal{A}_i$, the max of these numbers is required. This heuristic comes from the properties of the so called "dichotomic cut" that are analysed later in Section 5.4.3.

5.4.2 Pseudo-code of the A* Algorithm for Global FMSO Selection

According to the above specifications, the A* algorithm for global FMSO selection is given in Algorithm 5.1 that we call *Global A* Algorithm*.

Line 2 creates the CLOSED list that contains the set of global FMSO sets already evaluated. Line 3 creates the OPEN list that contains the global FMSO sets to evaluate, initially only the node n_0 is known. In line 5, the set cameFrom contains the best previous global FMSO sets. The score of the evaluation function is then computed. In line 6, the score g is calculated for n_0, in the first loop, $f(n_0) = h(n_0)$ because $g(n_0) = 0$. The h score is computed Line 8. Then a while loop determines first the current node with the lowest f score in the OPEN list. Next, we verify if the current node is equal to the target node: if it is the case, it means that the goal has been reached and the loop ends; if this is not the case, the current node of the OPEN list is moved to the CLOSED list and all possible neighbors are considered, knowing that a neighbor is an FMSO that increases the isolability degree. In line 24 the f score is computed for every neighbor. Finally, the optimum path is reconstructed with all the elements included in the CLOSED list.

In the following, we use a small academic system to illustrate the algorithm.

Example 5.4.1. *Assume that the decentralized diagnoser design procedure has provided 4 global FMSO sets along the hierarchy and that 5 faults are considered. The corresponding fault signature matrix is the one in Table 5.1.*

Algorithm 5.1: Global A* Algorithm.

```
1  Function Global A*(n_0,goal)
2      closedSet := {};
3      openSet := n_0;
4      cameFrom := the empty map;
5      g:= map with default value of Infinity;
6      g[n_0] := 0 ;
7      f := map with default value of Infinity;
8      f[n_0] := h(n_0, goal);
9      while openSet is not empty do
10         n_i := the node in openSet having the lowest f value;
11         if n_i = goal then
12             return reconstructpath(cameFrom, n_i);
13         else
14             openSet.Remove(n_i);
15             closedSet.Add(n_i);
16             for each neighbor of n_i do
17                 if neighbor not in closedSet then
18                     tentativeg := g[n_i] + 1;
19                 if neighbor not in openSet then
20                     openSet.Add(neighbor);
21                 else if tentativeg < g[neighbor] then
22                     cameFrom[neighbor] := n_i;
23                     g[neighbor] := tentativeg;
24                     f[neighbor] := g[neighbor] + h(neighbor, goal);
25     return failure
```

Φ	Faults				
	f_1	f_2	f_3	f_4	f_5
φ_1	0	0	0	0	1
φ_2	0	0	1	1	0
φ_3	0	1	0	1	0
φ_4	1	0	0	0	1

Table 5.1: Fault signature matrix of Example 5.4.1.

Figures 5.1 to 5.5 illustrate the nodes expanded by the A* Algorithm during the global FMSO selection.

First, the algoritm creates the *start node* n_0 as the state $\mathcal{A}_0 = \mathcal{F} = \{\{f_0, f_1, f_2, f_3, f_4, f_5\}\}$ with the whole set of faults with only one ambiguity set as shown in Figure 5.1. This node becomes the *current node*. This node is stored in the *Open List*.

Figure 5.1: *Node Start* for example 5.4.1.

Then, all the neighbors of the *current node* are determined, such as the nodes capable of cutting the ambiguity of this node of some degree. For each one of them the gscore and the hscore are calculated as given in Figure 5.2. All these neighbors are stored in the *Open List* and the start node is assigned as parent node of them.

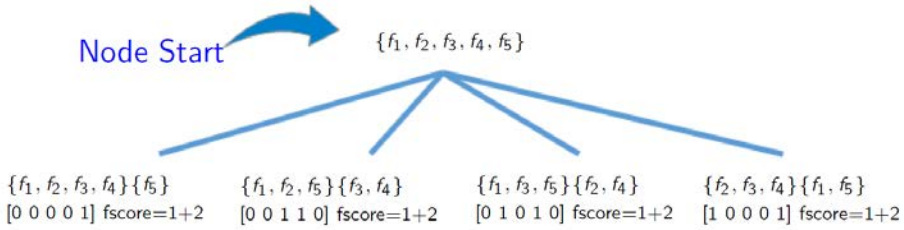


Figure 5.2: Expanded nodes during A* Algorithm for example 5.4.1.

Next, within the *Open List*, the node with the lowest value of **fscore = gscore + hscore** is selected, this node is moved to the *Closed List* and becomes the *current node*. In Figure 5.3, the FMSO set with signature matrix [00001] is selected, it has the minimum fscore because it produces two sets of ambiguity (when two or more sets have the same fscore, the first one that is in the *open list* is selected and if it is not the most suitable the algorithm will return and select another one). For the neighbors of the *current node* that are not in the closed list, a tentative gscore is calculated with the objective of determining if the path that is being followed is indeed adequate and does not need to return to find another path through the parent nodes.

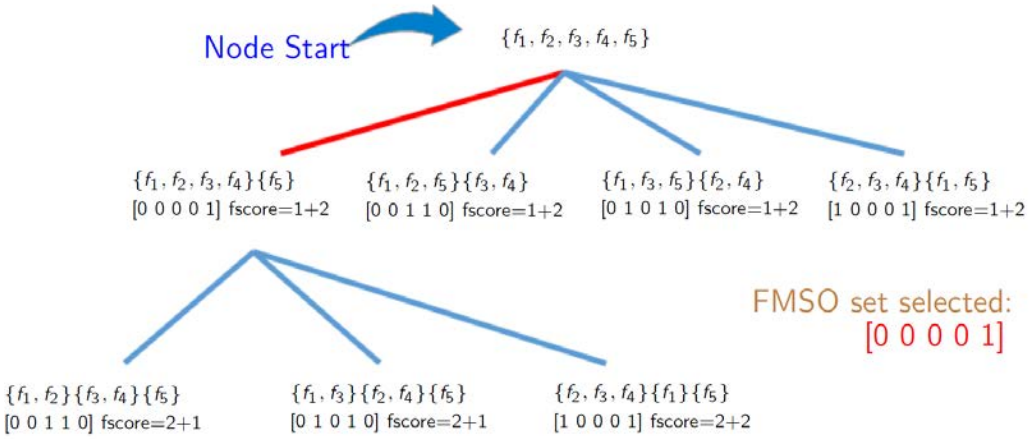


Figure 5.3: Expanded nodes during A* Algorithm for example 5.4.1.

Then, again all the neighbors of the *current node* are determined, such as the nodes capable of cutting the ambiguity of this node of some degree. In Figure 5.4, the FMSO set with signature matrix [00110] is selected, it has the minimum fscore because it produces three sets of ambiguity.

This loop repeats itself while not reaching the *goal node*, which refers to state $\mathcal{A}_f = \{\{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_5\}\}$., the algorithm continues to increase FMSO sets as shown in Figure 5.4 and 5.5

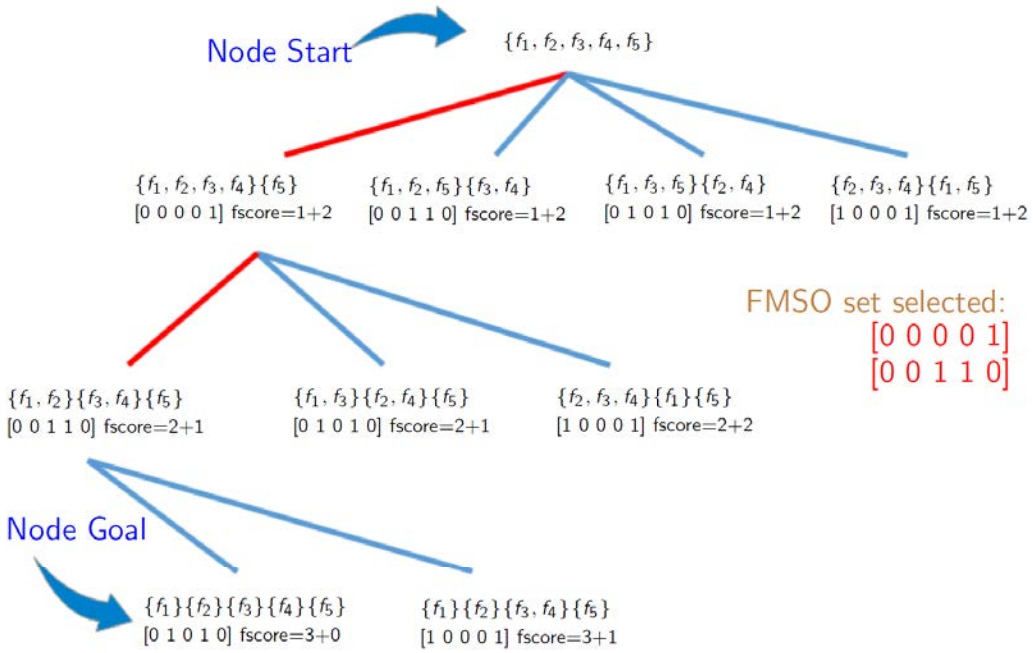


Figure 5.4: Expanded nodes during A* Algorithm for example 5.4.1.

Finally, as shown in Figure 5.5, the path is indicated by bold red edges in the corresponding graph and to achieve detectability and complete isolability for this system, 3 FMSO sets are required: $\Phi = \varphi_1, \varphi_2, \varphi_3$, which is indeed easily confirmed by Table 5.1.

5.4.3 Dichotomic Cut

The following property can be shown.

Property 6. *The isolability degree is a strictly increasing monotonic function.*

Proof. The isolability degree is the cardinal of the set $\mathcal{A}_i$, the set of ambiguity sets. Suppose that $\mathcal{I}_{S_i} = k$ and that a new global FMSO φ is chosen and denote f_φ its fault support. Three cases may occur for $\mathcal{A}_{i+1}$:

1. all faults in $\mathcal{A}_i^j, j = 1, \ldots, k$ are included in f_φ, so $Card(\mathcal{A}_{i+1}) = Card(\mathcal{A}_i)$ and the isolability degree remains unchanged.

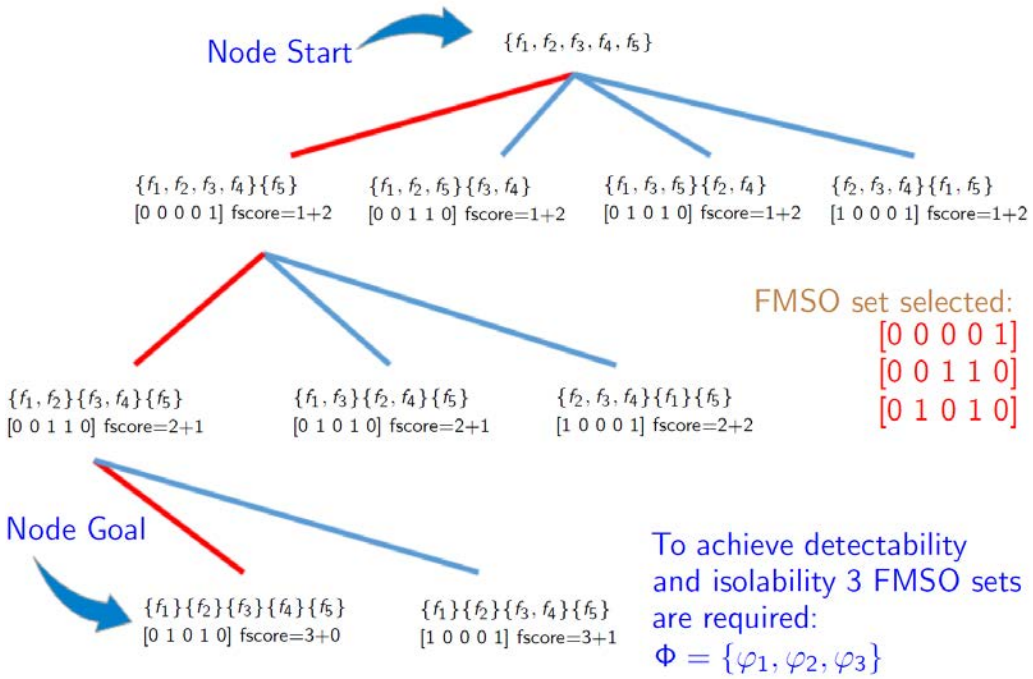


Figure 5.5: Detectability and complete isolability for example 5.4.1.

2. no fault of $\mathcal{A}_i^j, j = 1, \ldots, k$ is included in f_φ, so $Card(\mathcal{A}_{i+1}) = Card(\mathcal{A}_i)$ and the isolability degree also remains unchanged.

3. for some j's, some faults of $\mathcal{A}_i^j$ are included in f_φ and some are not. It means that f_φ partitions $\mathcal{A}_i^j$ in two parts and so $Card(\mathcal{A}_{i+1}) \geq Card(\mathcal{A}_i) + \nu$, where $1 \leq \nu \leq k$. Therefore the isolability degree increases by at least 1 and at most by k, i.e. it doubles.

Each time a global FMSO is chosen and added, the cardinal of the set $\mathcal{A}_i$ increases. It can increase from 0 to double. Therefore the isolability degree is a strictly increasing monotonic function. In conclusion, in cases 1 and 2, the isolability degree remains the same, i.e. $\mathcal{I}_{S_{i+1}} = \mathcal{I}_{S_i}$. In case 3, the isolability degree increases at least by 1 and maximum k, i.e. doubling size. Isolability is hence an increasing monotonic function. □

Property 7. *The dichotomic cut is the most efficient manner to increase the isolability degree.*

Proof. The dichotomic cut is a specific choice for a new global FMSO φ that has the property to partition each ambiguity set of $\mathcal{A}_i$ by 2, so that the isolability degree is doubled. It has been proved that the isolability degree can increase from 0 to double when adding a new FMSO set, in consequence the dichotomic cut is the most efficient manner to increase the isolability degree. □

The Global A* Algorithm identifies and selects the minimal number of global FMSO sets that allow to reach detectability and isolability for all the faults of the

system. Once selected, it is possible to position each global FMSO in its corresponding subsystem and in this way a distributed diagnostic system is designed. However, the main limitation of this solution is that all global FMSO sets are considered to be available, which, as discussed in Chapter 2, is in some cases not possible due to functional, geographical or privacy constraints.

In the following section, two iterative algorithms are proposed that allow the development of distributed fault diagnosis systems based on information available only for each subsystem, minimizing the interaction between them.

5.5 Distributed Case: Shared FMSO Sets Selection

In this section, we present two algorithms to build a minimal set of global FMSO sets that guaranty maximal detectability and isolability, starting only from shared FMSO/CMSO sets available for each subsystem:

- The $FirstLocalThenComplete$ algorithm and
- The $IterativeFindAndComplete$ algorithm.

The objective of this optimization is to minimize the number of tests and the interactions between subsystems.

In order to reach this objective, some procedures are required to allow the construction of global FMSO sets as compound FMSO sets, from the shared FMSO sets that are selected by the algorithm.

First we define the COMPLETE(φ) procedure that is used for finding a compound FMSO set that is global (Algorithm 5.2).

Algorithm 5.2: COMPLETE function.

```
1  Function COMPLETE(SharedFMSOsetSTART, subsystem, isolability)
2      Φ^c := SharedFMSOsetSTART;
3      subsystemSet = ∅;
4      closedSet := ∅;
5      for each shared variable x^s ∈ SharedFMSOsetSTART do
6          sharedFMSOPossibleSets := INCLUDE(x^s, subsystem) \ closedSet;
7          bestSharedFMSO = BEST(sharedFMSOPossibleSets);
8          subsystemSet = subsystemSet ∪ system that includes
             bestSharedFMSO;
9          Φ^c := Φ^c ∪ bestSharedFMSO;
10         closedSet := closedSet ∪ bestSharedFMSO;
11     return [Φ^c, subsystemSet];
```

For the COMPLETE function two functions are required: INCLUDE function and BEST function. The INCLUDE function (Algorithm 5.3) finds all the shared FMSO sets that involve a shared variable x^s.

Algorithm 5.3: INCLUDE function.

```
1 Function INCLUDE(x^s, Σ)
2     S := ∅;
3     for each subsystem Σ' ≠ Σ do
4         for each shared FMSO set φ ∈ Σ' s.t x^s ∈ X^s_φ do
5             S := S ∪ φ;
6     return S;
```

Afterwards, the BEST function (Algorithm 5.4) chooses among a set of shared FMSO sets the one including the least numbers of shared variables.

Algorithm 5.4: BEST function.

```
1 Function BEST(SharedFMSOPossibleSets)
2     bestSharedFMSO := ∅;
3     for each shared FMSO set φ do
4         [φ,SharedVariables,NumSubs] = FindSVandNumSub (φ);
5     Sort [φ,SharedVariables,NumSubs] from lowest to highest according to
        the number of shared variables and if the sets have the same number of
        shared variables sort according to the number of subsystems involved.
        bestSharedFMSO= φ_1;
6     return bestSharedFMSO;
```

In algorithm 5.4, the function $FindSVandNumSub$ finds the shared variables and counts the number of subsystems involved in the respective φ. Then sort function sorts the FMSO sets φ rom lowest to highest according to the number of shared variables and if the sets have the same number of shared variables sort according to the number of subsystems involved.

5.5.1 First Find all Local Solutions, then Complete for Isolability

The $FirstLocalThenComplete$ algorithm (Algorithm 5.5) uses A^* for each subsystem, considering that each shared FMSO set can be completed and so that shared variables are known.

The first part (Lines 3-7) of Algorithm 5.5 consists in finding a set of shared FMSO sets for each subsystem and in verifying whether those sets satisfy the local isolability constraint.

At Line 8, the algorithm tries to complete the selected shared FMSO sets and to get global FMSO sets, first within the shared FMSO sets already selected for the other subsystems, then with the COMPLETE procedure.

Algorithm 5.5: First Local then Complete algorithm.

```
1  Function FirstLocalThenComplete
2      sharedFMSOsetCandidates = ∅;
3      for i = 1 ... n do
4          φ_start := findStartIn(sharedFMSOsetsof(Σ_i));
5          isolability : = FaultsSetOf(Σ_i);
6          sharedFMSOSet_i := GlobalA*(φ_start, isolability);
7          sharedFMSOsetCandidates =
             sharedFMSOsetCandidates ∪ sharedFMSOSet_i;
8      Try to complete every set in sharedFMSOsetCandidates to get
         globalFMSOsets;
9      if every set in sharedFMSOsetCandidates can be completed with sets
         included in sharedFMSOsetCandidates then
10         return globalFMSOsets;
11     else
12         for each φ that cannot be completed within
             sharedFMSOsetCandidates do
13             COMPLETE(φ);
14         if success then
15             return globalFMSOsets;
16         else
17             return fail;
18
```

5.5.2 Find and Complete Iteratively

The *IterativeFindAndComplete* algorithm (Algorithm 5.6) uses the A^* algorithm for one subsystem, then tries directly to complete the set of selected shared FMSO sets by invoking shared FMSO/CMSO sets the other subsystems.

In this procedure, the algorithm seeks to reach the diagnostic objectives of a subsystem and then iteratively passes to the following.

After processing one subsystem, the A^* algorithm is then used to select a set of shared FMSO sets on another subsystem, and so on until all subsystems have been processed.

Line 3 in Algorithm 5.6 , L_{Σ}^{open} represents an open list containing the list of the subsystems not already treated. While this list is not empty, its first subsystem is found and removed from it (Line 5).

Line 9 isolates one shared FMSO set in the subsystem Σ. Line 10, isolability represents the set of faults represented in Σ. The COMPLETE procedure is used to complete the shared FMSO sets selected by the GlobalA* algorithm on line 11.

Algorithm 5.6: Find and complete iteratively algorithm.

```
1  Function IterativeFindAndComplete
2      globalFMSOsets := ∅;
3      L_Σ^open = {Σ_1, ..., Σ_n};
4      while L_Σ^open ≠ ∅ do
5          Σ_init := FindAndRemoveFirstSSOf(L_Σ^open);
6          L_Σ := Σ_init;
7          while L_Σ ≠ ∅ do
8              Σ=: FindAndRemoveFirstSSOf(L_Σ);
9              φ_start := findStartIn(sharedFMSOsetsof(Σ));
10             isolability : = FaultsSetOf(Σ);
11             sharedFMSOSet := GlobalA*(φ_start, isolability);
12             [Φ^c, subsystemSet] =
                COMPLETE(sharedFMSOSet, Σ, isolability);
13             L_Σ := L_Σ ∪ subsystemSet;
14             globalFMSOsets := globalFMSOsets ∪ Φ^c;
15     return globalFMSOsets
16
```

5.6 Conclusion

This chapter presented the optimization algorithmic contributions of this thesis. Three algorithms based on the A* algorithm are proposed. The first algorithm starts from global FMSO sets and selects a minimal set of global FMSO sets in order to maximize the isolability degree. Dichotomic cut is proved to be the most efficient manner to increase the isolability degree at once, hence the proposed heuristic function. Then two algorithms propose solutions for optimized distributed structural diagnoser design. The algorithm *FirstLocalThenComplete* uses an A* for each subsystem and then completes the selected shared FMSO sets. The algorithm *IterativeFindAndComplete* uses an A* for one subsystem and completes the set of selected shared FMSO sets among other subsystems before processing another subsystem. The algorithm iterates on subsystems until all of them have been processed. These algorithms are implemented with Matlab and tested on one realistic case study in the next Chapter 6. Results are presented in the next chapter.

Part IV

Application to industrial process and systems

Chapter 6

Case of Study: Decentralized Diagnosis for an ADCS of a Satellite LEO

Contents

6.1 Introduction

Artificial satellites are systems orbiting the Earth and other planets in the Solar System launched into orbit using rockets. Artificial satellites, called in this thesis only as satellites, come in a variety of sizes, shapes, and purposes [Schmude 2012]. The mission of a particular satellite determines what specialized equipment that particular satellite must carry. A communications satellite, for example, contains a special device called a transponder, which allows that satellite to receive a radio frequency (RF) signal at one frequency and then retransmit that signal back to Earth at another frequency. A scientific satellite, like NASA's Chandra X-ray Observatory, has a special collection of instruments, which gather high-energy astrophysics data from celestial objects. Military satellites, like the Defense Support Program's missile detection and warning satellites, use very special information gathering instruments, such as a sensitive infrared telescope that can detect hostile

missile launches. A weather satellite as Nimbus spacecraft with important technology advances as the fact that they all flew in near-polar, sun-synchronous orbits around Earth [Bendick 1991, Schmude 2012]. To accomplish their respective missions, usually the artificial satellites are classified according to their orbit of work, within four altitude classifications:

- Low Earth orbit (LEO): Geocentric orbits ranging in altitude from 180 km - 2000 km.
- Medium Earth orbit (MEO): Geocentric orbits ranging in altitude from 2000 km - 35786 km.
- Geosynchronous Orbit (GEO): Geocentric circular orbit with an altitude of 35786 kilometres. The period of the orbit equals one sidereal day, coinciding with the rotation period of the Earth. The speed is approximately 3000 metres per second (9,800 ft/s).
- High Earth orbit (HEO): Geocentric orbits above the altitude of geosynchronous orbit 35786 km.

This chapter presents the application of the decentralized fault diagnosis method on the Attitude Determination and Control System of a Low Earth Orbit satellite. This work is an improvement of the work presented in [Chanthery 2016]. First section describes the low orbit satellites. Then section 6.2.1 recalls dynamics of the LEO satellite and Section 6.3 gives the structural model used in the following. Section 6.4 applies the decentralized method for diagnosis on this structural model.

6.2 Mathematical Modeling of a Low Earth Orbit Satellites

An example of a low earth orbit satellite is the ISS (International Space Station) that orbits at 400 km with a speed of 28 000 km/hour with time for one orbit about 90 minutes [Rycroft 2002]. The low-orbit satellite has some specific characteristics, most of them move 0.4 - 1.0°/s or the length of a fully extended fist in 10–25 s. They usually remain visible for at least 30s. They drift silently across the sky and do not leave a vapor trail. Satellites following nearly polar orbits often have a nearly constant brightness. Those moving from west to east (or east to west) will grow brighter or dimmer as a result of a changing phase. A few also tumble and, hence, undergo rapid brightness changes [Schmude 2012].

In Figure 6.1 is shown the low earth orbit satellite SPOT-7 that is an optical imaging satellite capable of imaging the Earth with a resolution of 1.5 meter panchromatic and 6 meter multispectral (blue, green, red, near-IR) and will offer imaging products to customers in defense, agriculture, deforestation, environmental monitoring, coastal surveillance, engineering, oil, gas and mining industries.

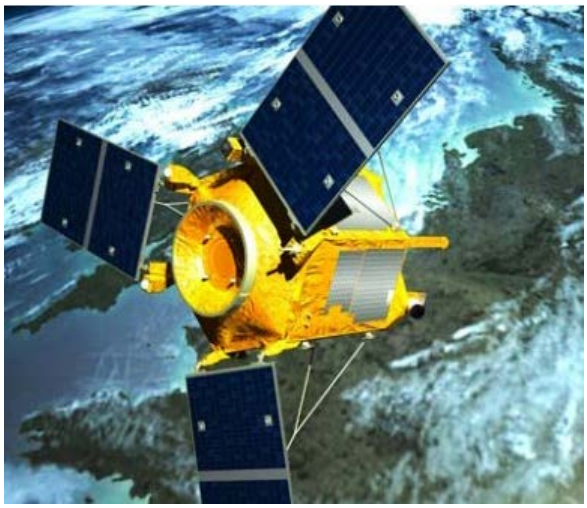


Figure 6.1: Spot 7 LEO satellite, source:http://www.intelligence-airbusds.com.

These satellites contain many instruments which must be oriented in a precise and specific direction. The main element studied in this work is the attitude determination and control system (ADCS) for which many solutions have been proposed. However, the fault diagnosis aspect remains an interesting and open subject. This system is composed by two subsystems: the attitude determination subsystem (ADS) and the attitude control subsystem (ACS). The composition of the ADCS of a typical satellite is represented in figure 6.2.

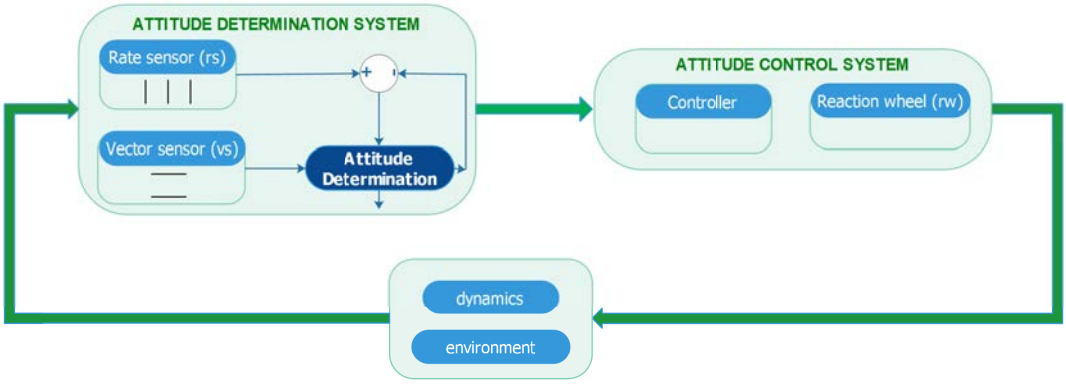


Figure 6.2: Attitude Determination and Control System of a typical satellite.

The attitude determination subsystem (ADS) is composed of sensors which sense the rate and angular position of the satellite. An attitude estimate is achieved using a sensor fusion (rate and vector sensors) [Pirmoradi 2009], which is provided as input to the attitude control subsystem (ACS). The ACS is composed of the control signal calculation and the actuators which provide the stabilizing and/or control torque to the satellite. The satellite under study is assumed to be a three-axis stabilized satellite in orbit around the earth. Here, reaction wheels and magnetorquers

are considered as actuators [Niemann 2003].

6.2.1 Dynamics of the Satellite

To analyze the motion of the satellite, two sets of coordinate systems are defined: a Sun-centered inertial frame with its origin at the center of the Sun and its third axis (z_i) normal to the ecliptic plane of the rotation of the Earth around the Sun (x_i, y_i) and a body-fixed frame which has its origin at the center of mass and its axes aligned with the principal axes of the satellite inertia.

The satellite is modeled as a rigid body having the moments of inertia matrix along the principal axes of rotation, I=Diag$_{3x3}\{I_x, I_y, I_z\}$. Assuming that x_b, y_b and z_b are the principal axes of inertia, the rotational motion of the satellite can be described in the body frame as follow [Pirmoradi 2009]:

$$\mathbf{I}\dot{w} = \mathbf{T} - w \times (\mathbf{I}w) \tag{6.1}$$

where the angular velocity vector w has components w_x, w_y and w_z, each along the body axes x_b, y_b and z_b of the satellite, $\dot{w}$ is the angular acceleration, and T is the total torque acting along the body axes.

The system of differential equations describing the vehicle attitude is:

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} 0 & s_\phi & c_\phi \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ c_\theta & s_\phi s_\theta & c_\phi s_\theta \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{6.2}$$

where ψ, θ and ϕ denote yaw, pitch and roll angles, respectively. The state vector of the satellite is $X = [\psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}]$. They are the angles by which the body frame is rotated relative to a reference frame.

6.2.2 Attitude Determination and Control System Modelling

The rate sensors of the satellite are three gyroscopes and the vector sensors are sun and star sensors. It is assumed that sensing axes of the rate gyros are aligned with each of the body axes of the satellite. The angular rate measurements from the gyros are used to solve the set of differential equations described by equation 6.2. w_x,w_y and w_z represent outputs of the three orthogonal rate gyros with their sensing axes aligned with the roll, pitch and yaw axes, respectively.

The attitude measurement from vector sensors is bounded and used to aid gyros to eliminate attitude drift error. The sensitive axes of the rate gyros are aligned with each of the body axes of the satellite. The modelling of the AD system is described in [Pirmoradi 2009]. The vector and rate sensor outputs are used to estimate the state vector both independently and merged together. These preliminary estimates are then fused together to arrive at the estimate which is feedback to the ACS. This redundancy can be used to check consistency.

The ACS is equipped with a three reaction wheels for 3-axis control. Another external torque source is necessary to unload the wheel's angular momentum. For

this satellite, magnetic torques are selected instead of thrusters that consume a large amount of fuel [Zuliana 2010].

6.2.3 Fault Scenarios

Faults are introduced in the system model equations. We consider faults in sensors and actuators specifically occurring in the rate and vector sensors of the ADS and in the reaction wheels of the ACS. Each of these faults has three components corresponding to the three axes. The faults are summarized in Table 6.1. Each of the faults can have three components corresponding to the three axis.

Table 6.1: Fault scenarios of the ADCS.

Component	Subsystem	Fault
Vector sensors (vs)	ADS	$f_{vs}(f_{vs_x}, f_{vs_y}, f_{vs_z})$
Rate sensors (rs)	ADS	$f_{rs}(f_{rs_x}, f_{rs_y}, f_{rs_z})$
Reaction wheel (rw)	ACS	$f_{rw}(f_{rw_x}, f_{rw_y}, f_{rw_z})$

6.2.4 Structural Model of the ADCS

The structure of the ADCS is abstracted as a set of constraints on a set of variables. Related information of such modelling can be founded in [Zuliana 2010, Pirmoradi 2009]. Most constraints are composed of three behavioral relations corresponding to three axes. From the set of variables of the system, the sensed quantities form the set of observed variables with all the rest assumed to be unobserved. The general procedure for the diagnoser design starts with assuming a small set of observed quantities, and can be optionally expanded to fulfill diagnosis and isolation capability specifications.

The global model $\Sigma(z, x, \mathtt{f})$ for this system is composed of 42 equations e_1 to e_{42} that relate the set of known variables Z, the set of unknown variables X and the set of system faults F, as presented below:

$$Z = \{\phi_{ref}, \theta_{ref}, \psi_{ref}, T_{cx}, T_{cy}, T_{cz}, W_{\omega_{xs}}, W_{\omega_{ys}}, W_{\omega_{zs}}, \dot{\psi}_s, \dot{\theta}_s, \dot{\phi}_s, \psi_s, \theta_s, \phi_s\}, \quad (6.3)$$

$$\begin{aligned} X = \{&T_x, T_y, T_z, RW_{am_x}, RW_{am_y}, RW_{am_z}, \dot{\psi}_{est}, \dot{\theta}_{est}, \dot{\phi}_{est}, \psi_{est}, \theta_{est}, \phi_{est}, \\ &dRW_{am_x}, dRW_{am_y}, dRW_{am_z}, RW_{\omega_x}, RW_{\omega_y}, RW_{\omega_z}, \dot{\psi}, \dot{\theta}, \dot{\phi}, \psi, \theta, \phi, \\ &\dot{\psi}_{est1}, \dot{\theta}_{est1}, \dot{\phi}_{est1}, \psi_{est1}, \theta_{est1}, \phi_{est1}, \dot{\psi}_{est2}, \dot{\theta}_{est2}, \dot{\phi}_{est2}, \psi_{est2}, \theta_{est2}, \\ &\phi_{est2}, \dot{\psi}_{est3}, \dot{\theta}_{est3}, \dot{\phi}_{est3}, \psi_{est3}, \theta_{est3}, \phi_{est3}\}, \end{aligned} \quad (6.4)$$

$$F = \{f_{vs_x}, f_{vs_y}, f_{vs_z}, f_{rs_x}, f_{rs_y}, f_{rs_z}, f_{rw_x}, f_{rw_y}, f_{rw_z}\} \quad (6.5)$$

The bi-adjacency matrix of the ADCS is shown in figure 6.3. The unobserved faults and observed variables are separated along the X-axis. The constraints that describe the behavior of the system components are described on the Y-axis. The structural model of the ADCS is composed of 42 constraints with 42 unobserved variables, 15 observed variables and 9 faults (modeled as variables in the bi-adjacency matrix).

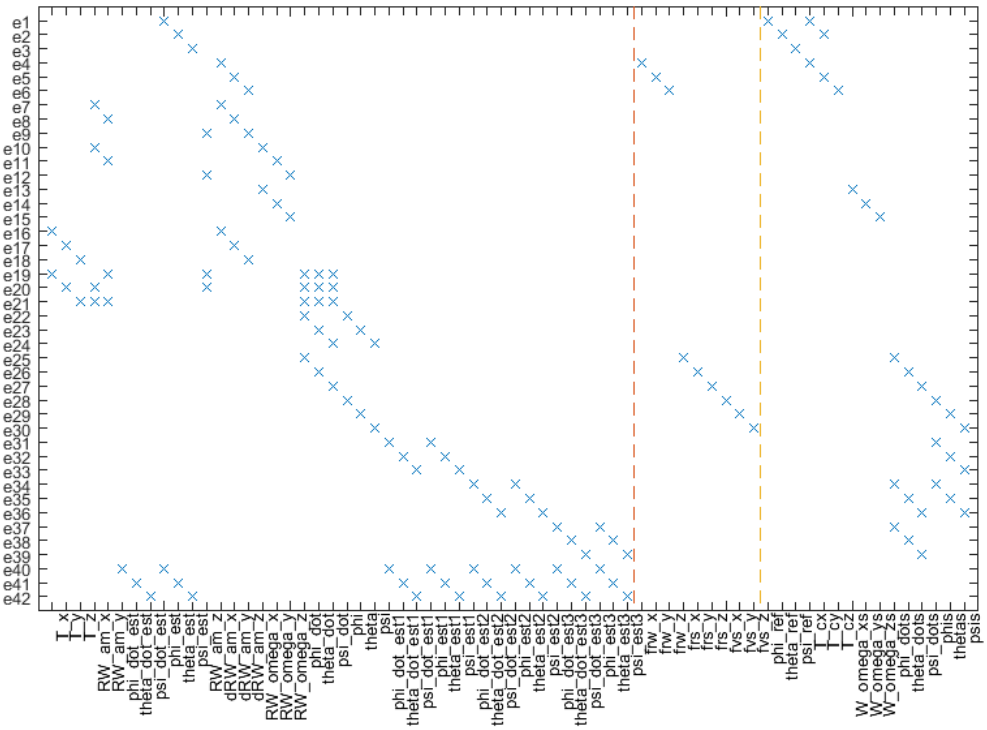


Figure 6.3: ADCS structure of a LEO satellite.

The structural model of the satellite ADCS, is considered to demonstrate the proposed decentralized architecture while still maintaining the same isolation capability power as the centralized approach and the advantageous isolation on request capability.

6.3 Decentralized Decomposition of the ADCS System of Satellite LEO

The natural decomposition of the ADCS system is an attitude control subsystem (ACS) as $\Sigma_{1,1}$ and an attitude determination subsystem (ADS) as $\Sigma_{1,2}$ as illustrated in Figure 6.4.

The equations representing the dynamics of the satellite (DYN) denoted $\Pi_{2,1}$ are considered as additional equations that are only available at level 2 for forming subsystem $\Sigma_{2,1}$. The restriction on the (DYN) equations may originate from differ-

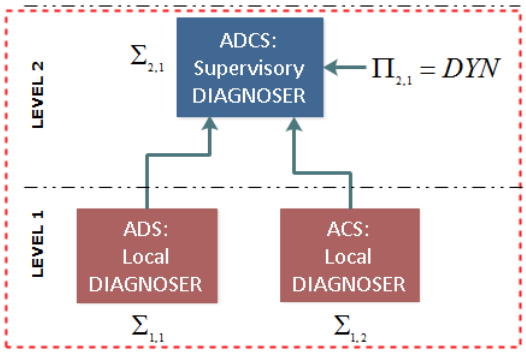


Figure 6.4: Architecture of the descentralized diagnoser designed for the ADCS system.

ent constraint types, e.g. confidentiality, distance and difficult access and they not therefore available at level 1.

The model decomposition of the ADCS system into subsystems $\Sigma_{1,1}$ and $\Sigma 1,2$ is given in Table 6.2.

Table 6.2: Model decomposition of the ADCS system into subsystems. $\Sigma_{1,i}$ $(i = 1, 2)$.

$$ACS = \Sigma_{1,1} = \begin{cases} \Sigma_{1,1} = & \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, ...e_{18}\} \\ F_{1,1} = & \{f_{rw_x}, f_{rw_y}, f_{rw_z}\} \\ X_{1,1} = & \{T_x, T_y, T_z, RW_{am_x}, RW_{am_y}, RW_{am_z}, dRW_{am_x}, dRW_{am_y}, \\ & dRW_{am_z}, RW_{\omega_x}, RW_{\omega_y}, RW_{\omega_z}\} \\ Z_{1,1} = & \{\phi_{ref}, \theta_{ref}, \psi_{ref}, T_{cx}, T_{cy}, T_{cz}, W_{\omega_{xs}}, W_{\omega_{ys}}, W_{\omega_{zs}}\} \end{cases}$$

$$ADS = \Sigma_{1,2} = \begin{cases} \Sigma_{1,2} = & \{e_{22}, e_{23}, e_{24}, e_{25}, e_{26}, e_{27}, e_{38}...e_{42}\} \\ F_{1,2} = & \{f_{vs_x}, f_{vs_y}, f_{vs_z}, f_{rs_x}, f_{rs_y}, f_{rs_z}\} \\ X_{1,2} = & \{T_x, T_y, T_z, RW_{am_x}, RW_{am_y}, RW_{am_z}, \dot{\psi}_{est}, \dot{\theta}_{est}, \dot{\phi}_{est}, \psi_{est}, \\ & \theta_{est}, \phi_{est}, \dot{\psi}, \dot{\theta}, \dot{\phi}, \psi, \theta, \phi, \dot{\psi}_{est1}, \dot{\theta}_{est1}, \dot{\phi}_{est1}, \psi_{est1}, \theta_{est1}, \\ & \phi_{est1}, \dot{\psi}_{est2}, \dot{\theta}_{est2}, \dot{\phi}_{est2}, \psi_{est2}, \theta_{est2}, \phi_{est2}, \dot{\psi}_{est3}, \dot{\theta}_{est3}, \dot{\phi}_{est3}, \\ & \psi_{est3}, \theta_{est3}, \phi_{est3}\} \\ Z_{1,2} = & \{\dot{\psi}_s, \dot{\theta}_s, \dot{\phi}_s, \psi_s, \theta_s, \phi_s\} \end{cases}$$

Additional equations referring to the subsystem $\Sigma_{2,1}$ are the following :

$$\Pi_{1,1} = \Pi_{1,2} = \{\emptyset\} \tag{6.6}$$

$$\Pi_{2,1} = \{e_{25}, e_{26}, e_{27}\} \tag{6.7}$$

The hierarchical decomposition of this system is shown in Figure 6.4, where in level 1, the 2 subsystems $\Sigma_{1,1}$ and $\Sigma_{1,2}$ are composed 21 equations each one , in level 2 the necessary information with additional 3 equations in $\Pi_{2,1}$ is grouped in subsystem $\Sigma_{2,1}$.

6.4 Decentralized Fault Diagnosis of the ADCS System of Satellite LEO

6.4.1 Global FMSO Sets Computation

As a reference, the global FMSO sets are computed for the whole ADCS system considered globally in order to determine maximal fault isolation.

ADCS system Global Diagnoser	
Max fault isolability	$[f_{vs_x}]$,$[f_{vs_y}]$,$[f_{vs_z}]$,$[f_{rs_x}]$,$[f_{rs_y}]$,$[f_{rs_z}]$,$[f_{rw_x}]$,$[f_{rw_y}]$,$[f_{rw_z}]$
FMSO sets	2448 FMSO sets

Table 6.3: FMSO sets for the Global System.

According to the results of Table 6.3, it can be seen that all faults can be detected and isolated with a centralized diagnoser for the ADCS system with 2448 FMSO sets.

6.4.2 Decentralized Diagnoser Design

Now, the Algorithm 3.1 (Section 3.2) of Chapter 3 for the decentralized diagnoser design is applied. We consider two levels $j = 1, 2$ with two subsystems $i = 1, 2$ for the first level $j = 1$ and one subsystem $i = 1$, for the second level ($j = 2$).

0.- As a previous step, with the information of Figure 6.3, it is possible to determine the vector of shared variables as:

$$X^s = \{T_x, T_y, T_z, \psi_{est}, \theta_{est}, \phi_{est}, RW_{am_x}, RW_{am_y}, RW_{am_z}\}, \tag{6.8}$$

These will be used to compute the *shared FMSO sets*.

1.- First, *local FMSO sets* are calculated for the subsystem $\Sigma_{1,1}$ of level 1 as given in Table 6.4.

$\Sigma_{1,1}$	
Max fault isolability	$[f_{rw_z}]$,$[f_{rw_y}]$,$[f_{rw_x}]$
Local FMSO sets $\Phi^l_{1,1} = \{\varphi_1, \varphi_2, \varphi_3\}$	
	$\varphi_1 = \{e_6, e_9, e_{12}, e_{15}\}$
	$\varphi_2 = \{e_5, e_8, e_{11}, e_{14}\}$
	$\varphi_3 = \{e_4, e_7, e_{10}, e_{13}\}$

Table 6.4: Subsystem $\Sigma_{1,1}$: $\Phi^l_{1,1}$.

As given in Table 6.4 it is possible to found detectability of faults f_{rw_x}, f_{rw_y} and f_{rw_z} with no additional information.

Next, for the case of the subsystem $\Sigma_{1,2}$ of level 1 *local FMSO sets* are calculated.

$$\Phi^{l}_{1,2} = \emptyset \tag{6.9}$$

The set of local FMSO sets for $\Sigma_{1,2}$ is empty. Hence, with no additional information, no fault can be diagnosed at level 1 for this subsystem.

2.- Next, for the subsystems $\Sigma_{1,1}$ and $\Sigma_{1,2}$, shared variables X^s are now assumed to be known and *shared FMSO sets* and *shared CMSO sets* are computed. Results are given in Table 6.5 and 6.6.

$\Sigma_{1,1}$

Max fault isolability	$[f_{rw_z}]$,$[f_{rw_y}]$,$[f_{rw_x}]$
Shared FMSO sets $\Phi^s_{1,1} = \{\varphi_4, \varphi_5, \varphi_6, \varphi_7, \varphi_8, \varphi_9\}$	
	$\varphi_4 = \{e_6, e_{18}\}$, $\varphi_5 = \{e_5, e_{17}\}$
	$\varphi_6 = \{e_5, e_8\}$, $\varphi_7 = \{e_4, e_{16}\}$
	$\varphi_8 = \{e_4, e_7\}$, $\varphi_9 = \{e_6, e_9\}$
Shared CMSO sets $\Psi^s_{1,1} = \{\psi_1, \psi_2, \psi_3, ..., \psi_9\}$	
	$\psi_1 = \{e_1\}$,$\psi_2 = \{e_2\}$, $\psi_3 = \{e_3\}$
	$\psi_4 = \{e_6, e_{17}\}$, $\psi_5 = \{e_8, e_{17}\}$
	$\psi_6 = \{e_9, e_{18}\}$, $\psi_7 = \{e_{10}, e_{13}\}$
	$\psi_8 = \{e_{11}, e_{14}\}$, $\psi_9 = \{e_{12}, e_{15}\}$

Table 6.5: Subsystem $\Sigma_{1,1}$: $\Phi^s_{1,1}$, $\Psi^s_{1,1}$.

$\Sigma_{1,2}$

Max fault isolability	$[f_{vs_x}]$,$[f_{vs_y}]$,$[f_{vs_z}]$
Shared FMSO sets $\Phi^s_{1,2} = \{\varphi_{10}, \varphi_{11}, \varphi_{12}..., \varphi_{24}\}$	
	$\varphi_{10} = \{e_{21}, e_{22}, e_{23}, e_{24}, e_{28}, e_{29}, e_{30}\}$
	$\varphi_{11} = \{e_{20}, e_{22}, e_{23}, e_{24}, e_{28}, e_{29}, e_{30}\}$
	$\varphi_{12} = \{e_{20}, e_{21}, e_{23}, e_{24}, e_{29}, e_{30}\}$
	$\varphi_{13} = \{e_{20}, e_{21}, e_{22}, e_{24}, e_{28}, e_{30}\}$
	$\varphi_{14} = \{e_{20}, e_{21}, e_{22}, e_{23}, e_{28}, e_{29}\}$
	$\varphi_{15} = \{e_{19}, e_{22}, e_{23}, e_{24}, e_{28}, e_{29}, e_{30}\}$
	$\varphi_{16} = \{e_{19}, e_{21}, e_{23}, e_{24}, e_{29}, e_{30}\}$
	$\varphi_{17} = \{e_{19}, e_{21}, e_{22}, e_{24}, e_{28}, e_{30}\}$
	$\varphi_{18} = \{e_{19}, e_{21}, e_{22}, e_{23}, e_{28}, e_{29}\}$
	$\varphi_{19} = \{e_{19}, e_{20}, e_{23}, e_{24}, e_{29}, e_{30}\}$
	$\varphi_{20} = \{e_{19}, e_{20}, e_{22}, e_{24}, e_{28}, e_{30}\}$
	$\varphi_{21} = \{e_{19}, e_{20}, e_{22}, e_{23}, e_{28}, e_{29}\}$
	$\varphi_{22} = \{e_{19}, e_{20}, e_{21}, e_{24}, e_{30}\}$
	$\varphi_{23} = \{e_{19}, e_{20}, e_{21}, e_{23}, e_{29}\}$
	$\varphi_{20} = \{e_{19}, e_{20}, e_{21}, e_{22}, e_{28}\}$
Shared CMSO sets $\Psi^s_{1,2} = \{\emptyset\}$	

Table 6.6: Subsystem $\Sigma_{1,2}$: $\Phi^s_{1,2}$, $\Psi^s_{1,2}$.

3.- Then, according to the algorithm, at level 2, the subsystem $\Sigma_{2,1}$ is considered.

For this purpose, shared FMSO and CMSO sets of the children systems of level 1 are considered together with the additional equations of $\Pi_{2,1}$ to form $\Sigma_{2,1}$ as shown in Equation 6.10.

$$\begin{aligned}\Sigma_{2,1} = \quad & \{e_1, e_2, ..., e_{24}, e_{28}, e_{29}, e_{30}\} \\ & \cup\{e_{25}, e_{26}, e_{27}\}\end{aligned} \tag{6.10}$$

Local FMSO sets are calculated for the subsystem $\Sigma_{2,1}$ at level 2 as given in Table 6.7.

$\Sigma_{1,1}$	
Max fault isolability	$[f_{rs_x}]$,$[f_{rs_y}]$,$[f_{rs_z}]$,$[f_{vs_x}]$,$[f_{vs_y}]$,$[f_{vs_z}]$
Local FMSO sets $\Phi^l_{2,1} = \{\varphi_{21}, \varphi_{22}...\varphi_{26}\}$	
	$\varphi_{21} = \{e_7, e_8, e_9, ..., e_{21}, e_{25}\}$
	$\varphi_{22} = \{e_7, e_8, e_9, ..., e_{21}, e_{26}\}$
	$\varphi_{23} = \{e_7, e_8, e_9, ..., e_{21}, e_{27}\}$
	$\varphi_{24} = \{e_7, e_8, e_9, ..., e_{21}, e_{22}, e_{28}\}$
	$\varphi_{25} = \{e_7, e_8, e_9, ..., e_{21}, e_{23}, e_{29}\}$
	$\varphi_{26} = \{e_7, e_8, e_9, ..., e_{21}, e_{24}, e_{30}\}$

Table 6.7: Subsystem $\Sigma_{2,1}$: $\Phi^l_{2,1}$.

4.- Based on the found local FMSO sets of $\Sigma_{2,1}$ given in Table 6.7, we conclude that it is possible to detect the nine faults of the system with analytical residual generators of level 2. However, one could use analytical residual generators obtained in a recursive way. Local analytical redundancy relations for subsystem $\Sigma_{1,1}$ can be used to isolate f_{rw_z}, f_{rw_y} and f_{rw_x} . Then, some of the six analytical redundancy relations computed from the local FMSO sets of $\Sigma_{2,1}$ can complete the isolation for the faults of the ADS f_{vs_x},f_{vs_y},f_{vs_z}, f_{rs_x},f_{rs_y} and f_{rs_z}. The isolation pattern is shown in the fault signature matrix of the Table 6.8.

	Faults								
	f_{rw_z}	f_{rw_y}	f_{rw_x}	f_{vs_x}	f_{vs_y}	f_{vs_z}	f_{rs_x}	f_{rs_y}	f_{rs_z}
$arr_1 \in ARR_{1,1}$	X								
$arr_2 \in ARR_{1,1}$		X							
$arr_3 \in ARR_{1,1}$			X						
$arr_4 \in ARR_{2,1}$				X					
$arr_5 \in ARR_{2,1}$					X				
$arr_6 \in ARR_{2,1}$						X			
$arr_7 \in ARR_{2,1}$							X		
$arr_8 \in ARR_{2,1}$								X	
$arr_9 \in ARR_{3,1}$									X

Table 6.8: FSM issued from the decentralized diagnoser for the ADCS system.

6.5 Conclusion

This chapter presents the application of the decentralized diagnosis for a real case study of Low Earth Satellite. The mathematical modelling of the LEO satellite is given. The structural model includes 42 equations and 42 unknown variables. 9 faults are considered to be of interest. A decentralized decomposition of the LEO satellite into 2 subsystems and one supervisory level is proposed. The computation of the set of global FMSO sets is done. 2448 global FMSO sets are found. Then the decentralized diagnoser design is tested. It proves that it is possible to detect and isolated 3 faults over 9 at the first level. The 6 other faults are isolated on the second level of the hierarchy.

Chapter 7

Case of Study: Distributed Diagnosis for a Reverse Osmosis Plant

Contents

7.1 Introduction

In response to increasing water scarcity, over the last 30 years desalination has evolved into a viable alternative water supply. It allows to tap non-traditional water resources with great potential to provide a sustainable, drought-proof water

supply. Desalination provides only around 1 percent of the world's drinking water, but this percentage is growing year-on-year. The use of seawater desalination plants using reverse osmosis methodology is currently a trend to meet the increasing requirements of drinking water around the world and its implementation presents a sustained growth in recent years being the most used methodology in this field [Voutchkov 2016].

According to the United Nations, close to 1.2 billion people already live in areas where freshwater is scarce. Another 1.6 billion people face chronic economic water shortage. While freshwater accounts for only 2.5% , seawater and brackish water found in oceans, seas and underground cover 97.5% of the total water in the world. Nowadays, obtaining this resource through reverse osmosis (RO) desalination method is an economically viable energy alternative [Dessouky 2002].

Currently, installing RO desalination plants is the trending, implying the implementation of increasingly large plants. These modern plants are sometimes distributed in large geographic areas, often implemented in different times and with different companies which involves the use of different equipment and technologies, often having restrictions on the confidentiality of information, e.g., internal technologies, mathematical models, etc.

Likewise, these systems are subject to different types of faults: in actuators such as high pressure pump, acid dosing pump, valves, and measures as flow, conductivity and temperature sensors; likewise common typical internal parameters faults of the membrane are weathering, fouling and scaling [Gambier 2009].

A Fault Tolerant Control (FTC) for a RO system based was developed by McFall [McFall 2007]; this control was based on physical switching logic reconfiguration by installing redundant control valves, which is a disadvantage due to the high implementation costs. The work of [Gambier 2009] presents a mathematical model of a laboratory reverse osmosis plant for a FDI system design considering faults in sensors, actuators and faults in RO system such as block of a pipeline, scaling/fouling and leaks. A FTC based on control loops reconfiguration was developed in this work, but not clear isolability analysis between considered faults was considered. [Garcia 2011] performed a monitoring fault detection system based on principal component analysis (PCA) technique for a simulated RO desalination plant. This technique allows to detect faults like offsets in pressure, temperature and concentration sensor, also blockages in filters and breakages in the membranes; but does not present faults in actuators like high pressure pump and acid dosing pump, and neither a fault isolation system is presented. Palacin [Palacin 2011] presented an enhanced dynamic library of reverse osmosis plants (ROSIM) used for simulation, optimization, fault detection and a simple fault tolerant control; however faults in actuators and sensors were not considered in the work.

The presented solutions solve the problem partially, allow to detect and isolate the most important faults of the systems of RO systems but do not consider the restrictions of the real plants: sustained growth, confidentiality of the information, geographical distribution, large amounts of computational processing to develop, many sensors and actuators to consider.

This chapter presents a solution proposal for the development of a fault diagnosis system with a distributed architecture considering subsystems and information limited to each one. The objective is to develop maximum diagnosability by calculating all necessary residual generators. The algorithms developed in the previous chapter, are executed to select the minimum necessary amount of residues to reach detectability and isolability for all the failures of interest.

7.2 Mathematical Modeling of the Reverse Osmosis Desalination Plant

The mathematical model developed here is based on the spiral wound reverse osmosis membrane configuration because it allows to seize the largest possible filter area compared with other configurations.

A component decomposition is necessary for obtaining a mathematical model of the RO system [Gambier 2009]. For the correct material balance, the system was divided into three components: membrane component, rejection component and permeate component. The considered variables are flows, pressures and concentrations. No energy balances were made because RO works at room temperature and there is not any phase change. Component decomposition is shown in Figure 7.1.

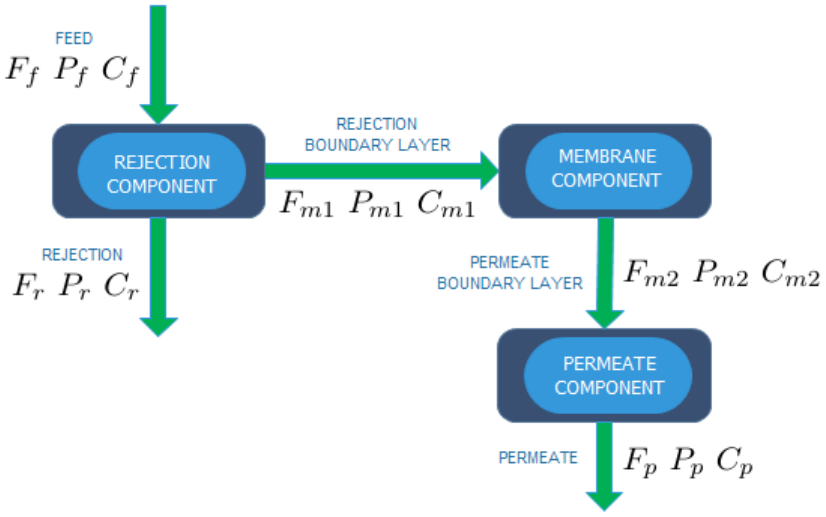


Figure 7.1: RO system decomposition.

7.2.1 Rejection Component

The global material balance for rejection component in Figure 7.1 is performed as follows:

$$dm_r/dt = F_f - F_{m_2} - F_r \tag{7.1}$$

Where m_r is the rejection holdup (kg), F_f is the feed flow water (kg/s), F_{m_2} is the permeate boundary layer flow (kg/s), F_r is the rejection flow water (kg/s). The partial material balance is given by the solute (salts) balance and is defined in differential form by:

$$d(C_r m_r)/dt = F_f C_f - F_p C_p - F_r C_r \tag{7.2}$$

Applying the partial derivative to equation (7.2) we obtain:

$$m_r(dC_r)/dt + C_r(dm_r)/dt = F_f C_f - F_p C_p - F_r C_r \tag{7.3}$$

Where C_r is the rejection concentration (kg/m^3), C_f is the feed water concentration (kg/m^3), C_p is the permeate water concentration (kg/m^3) and F_p is the permeate flow (Kg/s).

By replacing Equation 7.1 in Equation 7.3, we finally obtain the expression for the rejection concentration dynamic:

$$(dC_r)/dt = 1/m_r[F_f(C_f - C_r) - F_p(C_p - C_r)] \tag{7.4}$$

7.2.2 Permeate Component

The global material balance for permeate component in Figure 7.1 is performed as follows:

$$dm_p/dt = F_{m_2} - F_p \tag{7.5}$$

Where m_p is the permeate holdup (Kg). The partial material balance is given by the change of concentration for the mass in control holdup (permeate) over time and is defined by:

$$d(m_p C_p)/dt = F_{m_2} C_{m_2} - F_p C_p \tag{7.6}$$

Where C_{m_2} is the permeate boundary layer concentration (kg/m^3) and F_{m_2} is the permeate boundary layer flow.

Applying the partial derivative to Equation 7.6 we obtain:

$$m_p dC_p/dt + C_p dm_p/dt = F_{m_2} C_{m_2} - F_p C_p \tag{7.7}$$

By replacing Equation 7.5 in Equation 7.7, we finally obtain the expression for the permeate concentration dynamic:

$$dC_p/dt = (1/m_p)F_p(C_{m_2} - C_p) \tag{7.8}$$

7.2.3 Membrane Component

The pressure balance in the membrane component is given by:

$$P_f = P_r + P_p \tag{7.9}$$

Where P_f is the feed pressure, P_r is the rejection pressure and P_p is the permeate pressure, all of them in Pascal (Pa).

The set of equations that define the membrane component are given by the water transport equations and the salt transport equations.

The water transport equations are given by the following considerations. The Equation 7.10 defines the water flow rate through the semipermeable membrane [Senthilmurugan 2005].

$$F_p = J_v A \tag{7.10}$$

To calculate the permeate flow, the approach of Spiegler-Kedem-Katchalsky (SKK) [Ahmed 2013] is considered and is defined by:

$$F_p = AK_w(\Delta P - \sigma\Delta\pi) \tag{7.11}$$

Where A is the transfer area of the membrane, K_w is the water permeability coefficient and σ is osmotic pressure reflection coefficient.

Equation 7.11 shows that water flow rate through the membrane is proportional to the net differential pressure $\Delta P - \sigma\Delta\pi$. The hydraulic pressure drop ΔP through the membrane is determined by:

$$\Delta P = \frac{1}{2}(P_f + P_r) - P_p \tag{7.12}$$

The osmotic pressure drop $\Delta\pi$ throughout the membrane is linearly related with concentrations by the Van't Hoff relation [Jiang 2014]:

$$\Delta\pi = R_g T(C_{m_1} - C_p) \tag{7.13}$$

Where the concentrations C_{m_1} and C_p are molar concentrations in rejection boundary layer and permeate respectively. The expression for osmotic pressure drop in kg/m^3 is given by:

$$\Delta\pi = \frac{R_g T}{M_m}(C_{m_1} - C_p) \tag{7.14}$$

The salt flow through the membrane is defined by:

$$F_s = K_s A(C_{m_1} - C_p) \tag{7.15}$$

Where F_s is the permeate salt flow kg/s and K_s is the salt permeability coefficient m/s. Another expression for F_s is defined by:

$$F_s = F_p C_p \tag{7.16}$$

Expression for rejection boundary layer concentration of the membrane is given by [Gambier 2009]:

$$C_{m_1} = \frac{F_f C_f + (F_f - F_p)C_r}{(2F_f - F_p)} \tag{7.17}$$

Another expression for concentration C_{m_1} is given by:

$$C_{m_1} = C_{m_2}[1 + \frac{K_w}{K_s}(\Delta P - \sigma\Delta\pi)] \tag{7.18}$$

Where the expression for permeate boundary layer concentration of the membrane can be deduced as the expression given in:

$$C_{m_2} = \frac{C_f}{1 + \frac{K_w}{K_s}(\Delta P - \sigma\Delta\pi)(1 - \frac{F_p}{2F_f})} \tag{7.19}$$

The model can be simplified assuming that $C_{m_2} = C_p$ and $F_{m_2} = F_p$. This assumption will be considered in this work.

7.2.4 Additional Equations for RO System

7.2.4.1 Rejection factor

Another way to get an expression for permeate concentration C_p is given by equation in function of the rejection factor:

$$C_p = C_{m_1}(1 - R) \tag{7.20}$$

The expression for rejection factor is defined by:

$$R = \frac{(1 - F)\sigma}{(1 - \sigma F)} \tag{7.21}$$

Where the flow parameter F is given by the exponential relation [Senthilmurugan 2005]:

$$F = e^{(-J_v(1-\sigma))/K_s} \tag{7.22}$$

Similarly another expression for obtaining permeate flow F_p in terms of rejection factor is given by:

$$F_p = K_s A \frac{R}{1 - R} \tag{7.23}$$

7.2.4.2 Concentration Polarization

During filtration, accumulation of solutes occurs on the surface of the membrane (Rejection boundary layer). This accumulation produces a layer of concentration which can be determined by the model of concentration polarization illustrated in Figure 7.2 [Ahmed 2013]. Fick's law is used to make the flow balance around the rejection boundary layer and is defined by:

$$J_v C_p = J_v C - D\frac{dC}{dx} \tag{7.24}$$

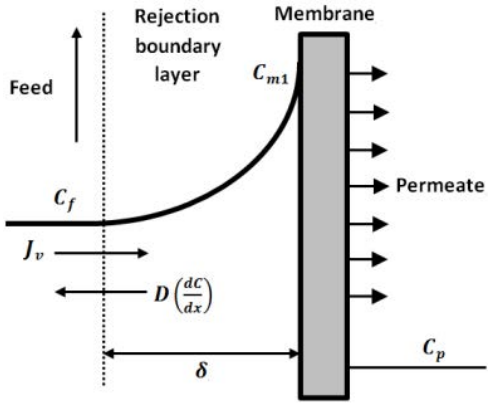


Figure 7.2: Concentration polarization model.

Expression for concentration polarization is obtained from Equation 7.24 and is given by [Khalaf 2008]:

$$\phi = \frac{C_{m_1} - C_p}{C_f - C_p} = e^{\frac{J_v \delta}{D}} \tag{7.25}$$

Where ϕ is the concentration polarization factor, δ is the boundary layer thickness (m), J_v is the permeate flow velocity (m/s), D is the solution diffusion coefficient and is defined by the following expression [Jiang 2014]:

$$D = 6.725(10^{-6})e^{(0.1546(10^{-3})C_f - \frac{2513}{273.15+T}} \tag{7.26}$$

Where C_f is the feed concentration and T is the feed solution temperature.

7.2.5 Relation Between pH and Conductivity

Expression that relates pH influence in permeate conductivity was taken from the experimental analysis made by Alatiqi [Alatiqi 1989] and is given by:

$$Cd_{pH} = -0.03626(pH_f - pH_i) \tag{7.27}$$

Where Cd_{pH} is conductivity due to pH change, pH_f is the final value of pH, pH_i is the initial value of pH. Equation 7.27 shows that a positive change in pH produces a reduction of the final permeate concentration and conversely. The final permeate concentration C_{ps} consists in the addition of concentration due to the feed pressure and the concentration due pH changes:

$$C_{ps} = C_p + C_{pH} \tag{7.28}$$

In practice, salt content is obtained by measuring conductivity uS/cm instead of concentration $kg/m3$. Kohlrausch equation relates conductivity (Cd) and concen-

tration C and is given by equation:

$$Cd = [126.45 - 0.3692523\sqrt{0.001C}](C/58440) \tag{7.29}$$

7.2.6 Adaptive Expert Generalized Predictive Multivariable Control System

The effective operation of these plants is not a simple task. RO desalination plants require accurate control system to increase the throughout and also to maintain operations close to the optimum conditions. For these requirements, an adaptive expert generalized predictive multivariable control (AEGPMC) was developed for the RO system under study. The development of this advanced control system is presented in detail in [Rivas-Perez 2017]. In this control system, were considered the permeate flow rate F_p and the permeate conductivity C_p as the output variables, and the feed pressure P_f and the brine flow rate F_r as the control variables. A scheme of the plant and the control system are shown in Figure 7.3.

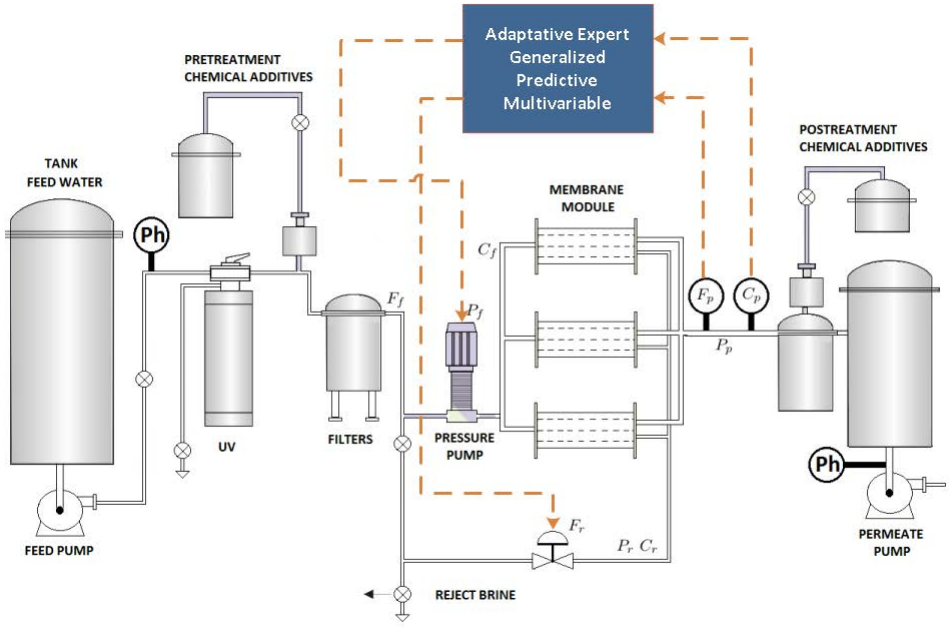


Figure 7.3: RO system with AEGPMC system.

For the AEGPMC, the optimal control law can be determined from minimizing the following quadratic cost function:

$$J = \sum_{j=N_1}^{N_2} \|\hat{y}(k+j) - r(k+j)\|_R^2 + \sum_{j=1}^{N_u} \|\Delta u(k+j-1)\|_Q^2 \tag{7.30}$$

where $\hat{y}(k+j)$ is an optimal j-step ahead prediction of the system output, $r(k+j)$ is the reference trajectory, N_1 and N_2 are the minimum and maximum

prediction horizons respectively, N_u is the control horizon, R and Q are positive definite weighting matrices. The control law is determined by solving the Equation 7.30 by:

$$\frac{\partial J}{dt} = 0 \tag{7.31}$$

Also, in terms of structural analysis, in Equation 7.30, J is a function of the permeate flow rate F_p, the permeate conductivity C_p, the feed pressure P_f and the brine flow rate F_r.

7.2.7 Equations for the RO System

According to the previous subsections, we can summarize the dynamic behavior of the RO system as a set of 25 equations, as shown in Table 7.1.

Figure 7.4 shows simulations for the dynamic of flows and concentrations. The simulation consider a constant value of feed pressure, where we note that flows have faster dynamic than concentrations. We can also see that brine concentration is greater than feed concentration and much higher than permeate concentration as it was expected. Boundary layer concentration is greater than feed concentration as we saw in Figure 7.2.

The structural representation of the RO system is given in Table 7.2.

7.2.8 Faults of Interest

A suitable set of faults $\boldsymbol{F}$ to describe the state of the most important faults in main elements of the reverse osmosis system is defined. Fault vector $\boldsymbol{F}$ is given by the following expression:

$$\boldsymbol{F} = (f_1, f_2, f_3, f_4, f_5, f_6, f_7) \tag{7.32}$$

Where f_1 is the conductivity sensor fault, f_2 is the flow sensor fault , f_3 is high pressure pump fault, f_4 is the temperature sensor fault, f_5 is the membrane scaling fault, f_6 is the membrane weathering fault and f_7 is the acid pump fault (pH pump).

The structural representation of the RO system with the set of faults of interest is given in Table 7.3.

7.3 Distributed Decomposition of the RO System

A decomposition of the RO system Σ into 3 subsystems Σ_i, $(i = 1..3)$ is defined as a partition of its equations (Section 2.4 , Chapter 2).

The first subsystem $\Sigma_1(z_1, x_1, \boldsymbol{f}_1)$ is described by the set of equations $\Sigma_1 = \{e_1, e_2, e_3, e_4, e_6, e_7, e_{23}, e_{25}\}$.

$X_1 = \{\Delta_p, \Delta_\pi\}$ is the set of subsystem unknown variables, $Z_1 = \{mF_p, mT\}$ is the set of subsystem measurements and $F_1 = \{f_1, f_5, f_6\}$ is the set of faults for this subsystem. The structural representation of subsystem Σ_1 is given in Table 7.3.

Table 7.1: Equations for the RO system.

e_1 : $\frac{dC_r}{dt} = \frac{1}{m_r}[F_f(C_f - C_r) - F_p(C_p - C_r)]$

e_2 : $\Delta P = 0.5(P_f + P_r) - P_p$

e_3 : $\Delta\pi = \frac{R_g T}{M_m}(C_{m_1} - C_p)$

e_4 : $J_v = K_w(\Delta P - \sigma\Delta\pi)$

e_5 : $F_p = J_v A$

e_6 : $C_p = \frac{C_f}{1 + \frac{K_w}{K_s}(\Delta P - \sigma\Delta\pi)(1 - \frac{F_p}{2F_f})}$

e_7 : $C_{m_1} = C_p[1 + \frac{K_w}{K_s}(\Delta P - \sigma\Delta\pi)]$

e_8 : $F_s = K_s A(C_{m_1} - C_p)$

e_9 : $F_p = \frac{F_s}{C_p}$

e_{10} : $F = e^{(-J_v(1-\sigma))/K_s}$

e_{11} : $R = \frac{(1-F)\sigma}{(1-\sigma F)}$

e_{12} : $C_p = C_{m_1}(1 - R)$

e_{13} : $C_{ps} = C_p + C_{pH}$

e_{14} : $C_{pH} = -0.03626(pH - 7)$

e_{15} : $C_{m_1} = \frac{F_f C_f + (F_f - F_p)C_r}{2F_f - F_p}$

e_{16} : $\phi = \frac{C_{m_1} - C_p}{C_f - C_p}$

e_{17} : $\phi = e^{\frac{J_v \delta}{D}}$

e_{18} : $D = 6.725(10^{-6})e^{(0.1546(10^{-3})C_f - \frac{2513}{273.15 + T_f}}$

e_{19} : $F_p = K_s A\frac{R}{1-R}$

e_{20} : $J = \sum_{j=N_1}^{N_2} \|\hat{y}(k+j) - r(k+j)\|_R^2 + \sum_{j=1}^{N_u} \|\Delta u(k+j-1)\|_Q^2$

e_{21} : $mP_f = P_f$

e_{22} : $mpH = pH$

e_{23} : $mF_p = F_p$

e_{24} : $mC_{ps} = C_{ps}$

e_{25} : $mT = T_f$

Similarly, The second subsystem $\Sigma_2(z_2, x_2, \boldsymbol{f}_2)$ is described by the set of equations $\Sigma_2 = \{e_5, e_8, e_9, e_{10}, e_{11}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{24}\}$.

$X_2 = \{F_s, F\}$ is the set of subsystem unknown variables, $Z_2 = \{mC_{ps}, mP_f\}$ is the set of subsystem measurements and $F_2 = \{f_3, f_7\}$ is the set of faults for this subsystem. The structural representation of subsystem Σ_2 is given in Table 7.5.

The third subsystem $\Sigma_3(z_3, x_3, \boldsymbol{f}_3)$ is described by the set of equations $\Sigma_3 =$

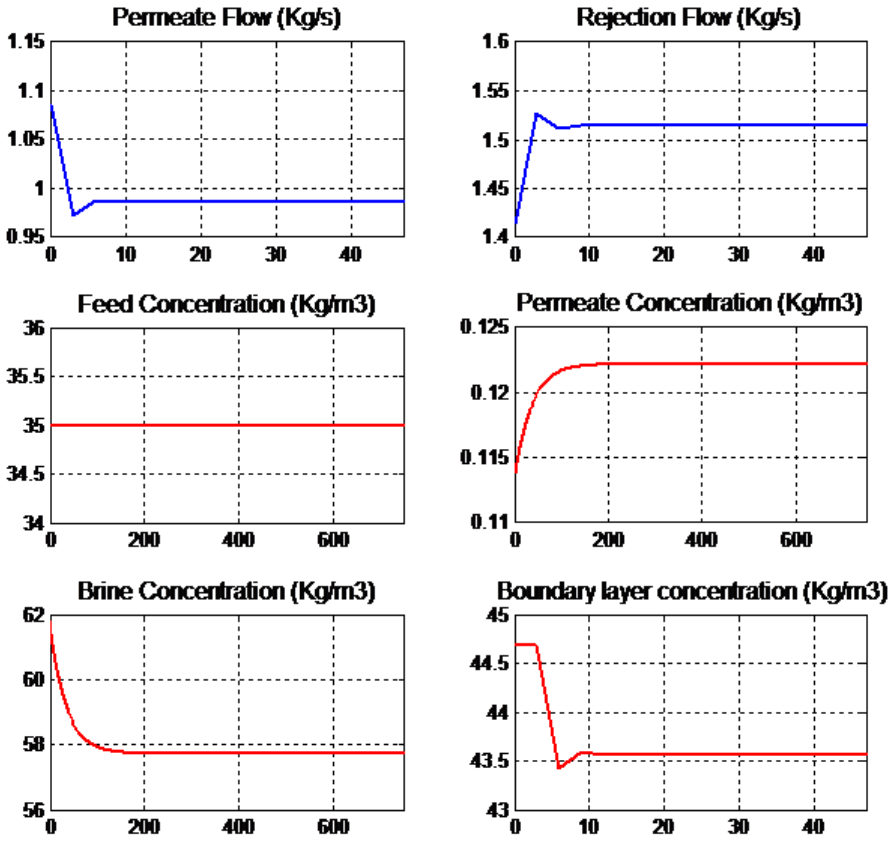


Figure 7.4: Plant Simulation Results.

$\{e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19}, e_{22}\}$.

$X_3 = \{C_{pH}\}$ is the set of subsystem unknown variables, $Z_3 = \{mpH\}$ is the set of subsystem measurements and $F_3 = \{f_4, f_2\}$ is the set of faults for this subsystem. The structural representation of subsystem Σ_2 is given in Table 7.6.

The RO system is highly coupled, it is observed that the set of shared variables has 12 unknown variables from a total of 17 unknown variables, $X^s = \{C_p, C_r, J_v, C_{m1}, R, pH, \phi, D, F_p, P_f, C_{ps}, T\}$. These shared variables are distributed as given in equations 7.33 to 7.36:

Eq	Unknown																Known				
	C_p	C_r	Δ_p	Δ_π	J_v	C_{m1}	F_s	F	R	C_{pH}	pH	ϕ	D	F_p	P_f	C_{ps}	T	mP_f	mpH	mF_p	mC_{ps}	mT
e_1	X	X												X								
e_2			X												X							
e_3	X			X		X											X					
e_4			X	X	X																	
e_5					X									X								
e_6	X		X	X										X								
e_7	X		X	X		X																
e_8	X					X	X															
e_9	X						X							X								
e_{10}					X			X														
e_{11}								X	X													
e_{12}	X					X			X													
e_{13}	X									X						X						
e_{14}										X	X											
e_{15}		X				X								X								
e_{16}	X					X						X										
e_{17}					X							X	X									
e_{18}													X				X					
e_{19}									X					X								
e_{20}											X			X	X	X						
e_{21}															X			X				
e_{22}											X								X			
e_{23}														X						X		
e_{24}																X					X	
e_{25}																	X					X

Table 7.2: Structural representation of RO system: unknown and known variables.

Eq	Unknown																Faults						
	C_p	C_r	Δ_p	Δ_π	J_v	C_{m1}	F_s	F	R	C_{pH}	pH	ϕ	D	F_p	P_f	C_{ps}	T	f_1	f_2	f_3	f_4	f_5	f_6	f_7
e_1	X	X												X										
e_2			X												X			X						
e_3	X			X		X											X					X		
e_4			X	X	X																		X	
e_5					X									X						X				
e_6	X		X	X										X										
e_7	X		X	X		X																		
e_8	X					X		X																
e_9	X							X						X										
e_{10}					X				X															
e_{11}									X		X													X
e_{12}	X					X				X														
e_{13}	X										X						X							
e_{14}											X		X						X					
e_{15}		X				X								X										
e_{16}	X					X							X											
e_{17}					X								X	X										
e_{18}														X			X				X			
e_{19}											X				X									
e_{20}														X	X	X								
e_{21}															X									
e_{22}														X										
e_{23}															X									
e_{24}																	X							
e_{25}																	X							

Table 7.3: Structural representation of RO system: unknown and faults of interest.

Eq	Unknown																	Known				
	C_p	C_r	Δ_p	Δ_π	J_v	C_{m1}	F_s	F	R	C_{pH}	pH	ϕ	D	F_p	P_f	C_{ps}	T	mP_f	mpH	mF_p	mC_{ps}	mT
e_1	X	X												X								
e_2			X												X							
e_3	X			X		X											X					
e_4			X	X	X																	
e_6	X		X	X										X								
e_7	X		X	X		X																
e_{23}														X						X		
e_{25}																	X					X

Table 7.4: Structural representation of subsystem Σ_1.

$$X^s_{1,2} = \{Jv, P_f\} \tag{7.33}$$

$$X^s_{1,3} = \{Cr, T\} \tag{7.34}$$

$$X^s_{2,3} = \{R, pH, \phi, D, C_{ps}\} \tag{7.35}$$

$$X^s_{1,2,3} = \{C_p, C_m, F_p\} \tag{7.36}$$

Eq	Unknown																Known				
	C_p	C_r	Δ_p	Δ_π	J_v	C_{m1}	F_s	F	R	C_{pH}	pH	ϕ	D	F_p	P_f	C_{ps}	T	mP_f	mpH	mF_p	mC_{ps}	mT
e_5					X									X								
e_8	X					X	X															
e_9	X						X							X								
e_{10}					X			X														
e_{11}								X	X													
e_{17}					X							X	X									
e_{20}											X			X	X	X						
e_{21}															X			X				
e_{24}																X					X	

Table 7.5: Structural representation of subsystem Σ_2.

Eq	Unknown																Known				
	C_p	C_r	Δ_p	Δ_π	J_v	C_{m1}	F_s	F	R	C_{pH}	pH	ϕ	D	F_p	P_f	C_{ps}	T	mP_f	mpH	mF_p	mC_{ps}	mT
e_{12}	X					X			X													
e_{13}	X									X						X						
e_{14}										X	X											
e_{15}		X				X								X								
e_{16}	X					X						X										
e_{18}													X				X					
e_{19}									X					X								
e_{22}											X								X			

Table 7.6: Structural representation of subsystem Σ_3.

7.4 Distributed Fault Diagnosis of RO System

7.4.1 Compute the Set of Global FMSO Sets Following the Distributed Approach

Running the algorithm 4.1, we calculate local FMSO sets Φ_i^l, shared FMSO sets Φ_i^s and shared CMSO sets Ψ_i^s of each subsystem ($i = 1..3$). The results are the following. Local FMSO sets are given by:

$$\Phi_1^l = \Phi_2^l = \Phi_3^l = \{\emptyset\}. \tag{7.37}$$

With no additional measurements, no fault can be diagnosed, as in Σ_1, Σ_2 and Σ_3 no local FMSO sets were found.

Shared FMSO sets and shared CMSO sets are given by Tables 7.7, 7.8 and 7.9. They are computed assuming that shared variables are known.

Then with each shared FMSO set as root FMSO set, we found all the compound FMSO sets $\varphi \in \Phi^c$ for the RO system as if a global model is not available: Algorithm 4.1 obtain the whole set of global FMSO sets (5173 FMSO sets), which means the same analytical redundancy relations while maintaining the confidentiality of each local model.

7.4.2 Distributed Generation of an Optimized set of Global FMSO Sets using the Algorithm LD

As it is observed in the previous section, a significant quantity of Global FMSO sets was found, therefore it is pertinent to build only the compound FMSO sets that guarantee to reach the detectability and isolation necessary for all the faults of interest. We build the local diagnosers with these specifications, first using the Algorithm LD (Algorithm 4.2 in Chapter 4) to select the optimal FMSO sets with

Σ_1	
Max fault isolability	$[f_1], [f_5], [f_6]$
Shared FMSO sets $\Phi_1^s = \{\varphi_1, \varphi_2 ... \varphi_{10}\}$	
	$\varphi_1 = \{e_4, e_6, e_7\}$
	$\varphi_2 = \{e_3, e_6, e_7\}$
	$\varphi_3 = \{e_3, e_4, e_7\}$
	$\varphi_4 = \{e_3, e_4, e_6\}$
	$\varphi_5 = \{e_2, e_6, e_7\}$
	$\varphi_6 = \{e_2, e_4, e_7\}$
	$\varphi_7 = \{e_2, e_4, e_6\}$
	$\varphi_8 = \{e_2, e_3, e_7\}$
	$\varphi_9 = \{e_2, e_3, e_6\}$
	$\varphi_{10} = \{e_2, e_3, e_4\}$
Shared CMSO sets $\Psi_1^s = \{\psi_1, \psi_2, \psi_3\}$	
	$\psi_1 = \{e_{23}\}$
	$\psi_2 = \{e_{25}\}$
	$\psi_3 = \{e_1\}$

Table 7.7: Subsystem Σ_1: Φ_1^s, Ψ_1^s.

Σ_2	
Max fault isolability	$[f_5], [f_7]$
Shared FMSO sets $\Phi_2^s = \{\varphi_{11}, \varphi_{12}, \varphi_{13}\}$	
	$\varphi_{11} = \{e_5, e_8, e_9\}$
	$\varphi_{12} = \{e_5, e_{20}\}$
	$\varphi_{13} = \{e_{11}\}$
Shared CMSO sets $\Psi_2^s = \{\psi_4, ..., \psi_8\}$	
	$\psi_4 = \{e_{10}\}$
	$\psi_5 = \{e_{17}\}$
	$\psi_6 = \{e_{21}\}$
	$\psi_7 = \{e_{24}\}$
	$\psi_8 = \{e_8, e_9, e_{20}\}$

Table 7.8: Subsystem Σ_2: Φ_2^s, Ψ_2^s.

the minimum amount of information (known variables) among the 3 subsystems.

According to algorithm LD, 7 shared FMSO sets are selected (one for each fault). Considered as root FMSO sets they are completed and give rise to compound FMSO sets as shown in Tables 7.10 and 7.11.

Based on the found compound FMSO sets, it can be seen that it is possible to detect the 7 faults of the system starting only with the information of each subsystem. Then, 7 analytical redundancy relations (ARR_i, $i = 1...7$) are computed from these 7 compound FMSO sets using an analytical residual generator calculation block. Finally the isolation founded is shown in the fault signature matrix of the Table 7.12. These results demonstrate that all considered faults can be isolated.

Σ_3	
Max fault isolability	$[f_2], [f_4]$
Shared FMSO sets $\Phi_3^s = \{\varphi_{14}, \varphi_{15}\}$	
	$\varphi_{14} = \{e_{13}, e_{14}\}$
	$\varphi_{15} = \{e_{18}\}$
Shared CMSO sets $\Psi_3^s = \{\psi_9, ..., \psi_{13}\}$	
	$\psi_9 = \{e_{12}\}$
	$\psi_{10} = \{e_{15}\}$
	$\psi_{11} = \{e_{16}\}$
	$\psi_{12} = \{e_{19}\}$
	$\psi_{13} = \{e_{22}\}$

Table 7.9: Subsystem Σ_3: Φ_3^s, Ψ_3^s.

Root FMSO sets $\Phi_7^r = \{\varphi_1^r, ..., \varphi_7^r\}$		
	$\varphi_1^r = \{e_2, e_6, e_7\}$	$F_{\varphi_1^r} = \{f_1\}$
	$\varphi_2^r = \{e_{13}, e_{14}\}$	$F_{\varphi_2^r} = \{f_2\}$
	$\varphi_3^r = \{e_5, e_{20}\}$	$F_{\varphi_3^r} = \{f_3\}$
	$\varphi_4^r = \{e_{18}\}$	$F_{\varphi_4^r} = \{f_4\}$
	$\varphi_5^r = \{e_3, e_6, e_7\}$	$F_{\varphi_5^r} = \{f_5\}$
	$\varphi_6^r = \{e_4, e_6, e_7\}$	$F_{\varphi_6^r} = \{f_6\}$
	$\varphi_7^r = \{e_{11}\}$	$F_{\varphi_7^r} = \{f_7\}$

Table 7.10: Selected root FMSO sets for the RO system.

Compound FMSO sets $\Phi^c = \{\varphi_1^c, ..., \varphi_7^c\}$		
	$\varphi_1^c = \{e_2, e_6, e_7, e_8, e_9, e_{12}, e_{19}, e_{21}, e_{23}\}$	$F_{\varphi_1^c} = \{f_1\}$
	$\varphi_2^c = \{e_8, e_9, e_{12}, e_{13}, e_{14}, e_{19}, e_{22}, e_{23}, e_{24}\}$	$F_{\varphi_2^c} = \{f_2\}$
	$\varphi_3^c = \{e_5, e_{10}, e_{11}, e_{19}, e_{20}, e_{21}, e_{22}, e_{24}\}$	$F_{\varphi_3^c} = \{f_3, f_7\}$
	$\varphi_4^c = \{e_5, e_8, e_9, e_{12}, e_{16}, e_{17}, e_{18}, e_{19}, e_{23}, e_{25}\}$	$F_{\varphi_4^c} = \{f_3, f_4\}$
	$\varphi_5^c = \{e_3, e_6, e_7, e_8, e_9, e_{12}, e_{19}, e_{23}, e_{25}\}$	$F_{\varphi_5^c} = \{f_5\}$
	$\varphi_6^c = \{e_4, e_5, e_6, e_7, e_8, e_9, e_{12}, e_{19}, e_{23}\}$	$F_{\varphi_6^c} = \{f_3, f_6\}$
	$\varphi_7^c = \{e_5, e_{10}, e_{11}, e_{19}, e_{23}\}$	$F_{\varphi_7^r} = \{f_3, f_7\}$

Table 7.11: Compound FMSO sets for the RO system.

7.4.3 Distributed Generation of an Optimized Selection of FMSO Sets with A* Algorithm

While Algorithm LD minimizes the communication between subsystems, this section will demonstrate that using the GlobalA* Algorithm, (Algorithm 5.2 in Chapter 5), it is possible to select a minimum set of FMSOs that also allows to comply with the detection and isolation specifications.

Based in the heuristic search framework, it is possible to determine a minimum

	Faults						
	f_1	f_2	f_3	f_4	f_5	f_6	f_7
ARR_1	X						
ARR_2		X					
ARR_3			X				X
ARR_4			X	X			
ARR_5					X		
ARR_6			X			X	
ARR_7			X				X

Table 7.12: Fault signature matrix of distributed diagnoser designed by the algorithm LD.

number of FMSOs to isolate all faults. The GlobalA* algorithm starts from an initial node with the state $\mathcal{A}_0 = \{\{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}\}$ that includes a unique set of ambiguity including all the faults of the system. The goal node has the state $\mathcal{A}_f = \{\{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_5\}, \{f_6\}, \{f_7\}\}$ (total isolability).

The GlobalA* algorithm finds that to detect and isolate the 7 faults of interest, it is enough to use 3 global FMSO sets as given in Table 7.13.

Global FMSO sets $\Phi = \{\varphi_1^c, \varphi_2^c, \varphi_3^c\}$		
	$\varphi_1 = \{e_1, e_8, e_9, e_{10}, e_{11}, ..., e_{18}, e_{20}, e_{21}, e_{22}, e_{25}\}$	$F_{\varphi_1} = \{f_2, f_4, f_7\}$
	$\varphi_2 = \{e_1, e_4, e_5, e_6, e_7, e_{13}, ..., e_{18}, e_{21}, e_{22}, e_{25}\}$	$F_{\varphi_2} = \{f_2, f_3, f_4, f_6\}$
	$\varphi_3 = \{e_1, e_3, e_5, e_6, e_7, e_8, e_9, e_{12}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}\}$	$F_{\varphi_3} = \{f_3, f_4, f_5\}$

Table 7.13: Optimal Global FMSO sets found by the GlobalA* algorithm.

The goal node $\mathcal{A}_f = \{\{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}, \{f_5\}, \{f_6\}, \{f_7\}\}$ is shown in the fault signature matrix on Table 7.14.

	Faults						
	f_1	f_2	f_3	f_4	f_5	f_6	f_7
$arr_1 \in ARR_1$		X		X			X
$arr_2 \in ARR_2$		X	X	X		X	
$arr_3 \in ARR_3$			X	X	X		

Table 7.14: Fault signature matrix of distributed diagnoser designed by the GlobalA* algorithm.

7.5 Conclusion

This chapter presents the application of the distributed diagnosis for a real case study of reverse osmosis desalination plant. The mathematical modeling of the

reverse osmosis system is given. The structural model includes 25 equations and 17 unknown variables. 7 faults are considered to be of interest.

A distributed decomposition of the RO system into 3 subsystems is proposed. This distributed system is highly coupled, as 12 unknown variables over 17 are shared variables.

The computation of the set of global FMSO sets following the distributed approach is done. 5173 global FMSO sets are found. Then distributed generation of an optimized set of global FMSO sets using the Algorithm LD is investigated. 7 global FMSO sets are built from 7 selected root FMSO sets. They achieve detectability and isolability of the 7 faults of interest. Finally the distributed generation of an optimized set of global FMSO sets using the GlobalA* algorithm is tested. 3 global FMSO sets are selected to achieve the detectability and isolability on the system, minimizing the number of selected global FMSO sets.

Part V

Conclusion and perspectives

Conclusions and Perspectives

Conclusions

Centralized architecture are sometimes not applicable for large-scale interconnected systems. They can be very expensive and lack robustness and adaptability. The distributed architecture is thus becoming a very challenging problem as systems become more and more complex. As a part of the architecture, the diagnosis function has to be distributed on different subsystems to remain efficient.

The motivation of this work is to develop efficient model-based decentralized and distributed diagnosis techniques to detect and isolate faults impacting system continuous dynamics. One objective of the work is to give a framework for future studies in distributed and decentralized fault diagnosis. Based on structural analysis, the diagnosis approaches that are proposed can advantageously handle non-linear and linear differential systems.

Chapter 1 recalls the framework of model-based diagnosis methods, especially the structural approach. The structural model of a system is an abstraction of its behaviour model that only keeps the structure, i.e. the existence of links between variables and constraints. From the diagnosis point of view, structural analysis leads to analytical redundancy based residuals. Chapter 1 introduces the concept of Fault-driven Minimal Structurally Overdetermined (FMSO) set that is an MSO set whose fault support is not empty. It combines the notion of MSO set and the one of fault support, in order to fault-focus the search on the main interesting sets of equations. These equations lead to the most interesting test sets. We also define Clear Minimal Structurally Overdetermined (CMSO) set that is MSO set whose fault support is empty.

Chapter 2 defines the two major notions of decentralized and distributed fault diagnosis architectures. Then, we propose to define the notion of subsystems in each case and reconsider the concepts of FMSO sets and CMSO sets in the decentralized and distributed cases. We thus define local variables, shared variables, local and shared FMSO/CMSO sets and finally compound and root FMSO sets. The contribution of this chapter is a proposition that states the conditions for which a union of shared FMSO/CMSO sets originating from different subsystems forms a global FMSO set also named compound FMSO set. A compound FMSO set has a specific AND/OR tree structure.

Chapter 3 deals with decentralized diagnosis via structural analysis. The main contribution of this chapter is the proposition of an algorithm for decentralized diagnoser design. The diagnoser is designed offline and takes into account the constraints induced by inter-level communication. The online implementation of the fault diagnosis system requires the definition of a fault signature matrix for each subsystem. The fault isolation process happens for each subsystem for which local FMSO sets have been computed based on their fault signature matrix. This

method is illustrated on a four-tank system.

Chapter 4 deals with distributed diagnosis via structural analysis. The main contribution of this chapter is the description of the operational procedure for the implementation of a distributed fault diagnosis system. A first algorithm generates the set of global FMSO sets only considering information about subsystems. The offline step computes analytical residual generators and the online step achieves fault detection with residual generator banks and a fault signature matrix for each subsystem. A second algorithm shows how to generate optimal local diagnosers for each subsystem forming compound FMSO sets based in the union of 'best' shared FMSO sets.

Chapter 5 first presents related work with respect to optimal test selection for fault diagnosis and solutions for the planning problem, specifically the A* algorithm. Some important notions and definitions as nodes, states, criteria and heuristics are presented. The main contribution of this chapter is the design of A*-like algorithms to solve the test selection problem in different contexts. The first algorithm proposes a solution for the centralized or decentralized architecture. It selects a minimal set of global FMSO sets of a system. The global FMSO sets may be generated in a centralized way or using the decentralized method proposed in Chapter 3: the set of global FMSO sets given as input is already reduced thanks to the isolation on request strategy. Then, two additional algorithms are presented to build global FMSO sets based on shared FMSO and CMSO sets. Algorithm $FirstLocalThenComplete$ uses an A* in parallel for each subsystem and after the selection, it tries to complete the selected shared FMSO sets. Algorithm $IterativeFindAndComplete$ uses an A* for one subsystem and tries to complete the set of selected shared FMSO sets invoking other subsystems. Then the algorithm processes iteratively all the subsystems.

In Chapter 6, the application of the decentralized fault diagnosis method proposed in Chapter 3 for a real case study of the Attitude Determination and Control System of a real case study of Low Earth Orbit satellite is presented. The mathematical modeling of the LEO satellite is given, next, the structural model that includes 42 equations and 42 unknown variables. 9 faults are considered to be of interest. A decentralized decomposition of the LEO satellite into 2 subsystems and one supervisory level is proposed. The computation of the set of global FMSO sets is done. 2448 global FMSO sets are found. Finally, the decentralized diagnoser design is tested. It proves that it is possible to detect and isolated 3 faults over 9 at the first level. The 6 other faults are isolated on the second level of the hierarchy.

In Chapter 7, the application of the distributed diagnosis method proposed in Chapter 4 for a real case study of Reverse Osmosis Desalination Plant is developped. The mathematical modeling of the reverse osmosis system is given, and next, the structural model that includes 25 equations and 17 unknown variables. 7 faults are considered to be of interest. A distributed decomposition of the RO system into 3 subsystems is proposed. This distributed system is highly coupled, as 12 unknown variables over 17 are shared variables. The computation of the set of global FMSO sets leads to 5173 global FMSO sets. Then the distributed generation of an

optimized set of global FMSO sets is investigated. 7 global FMSO sets are built from 7 selected root FMSO sets. They achieve detectability and isolability of the 7 faults of interest. Finally the distributed generation of an optimized set of global FMSO sets is done using an A*-based algorithm of Chapter 5. In conclusion, 3 global FMSO sets are selected to achieve full detectability and isolability, minimizing the number of selected global FMSO sets.

Perspectives

This work provides a framework for future studies in distributed and decentralized fault diagnosis and opens a large number of perspectives.

Chapter 3, 4 and 5 propose solutions to find the minimal set of tests in the architecture, or the minimal amount of communication between subsystems. One perspective is to generalize the optimization criterion and to propose **multi-criteria optimization for the test selection problem**. The goal could be to optimize the number of selected tests and the amount of communication, while maximizing the opportunity to fast detect and isolate the most critical faults. The algorithms should thus be adapted, as well as the heuristics, to solve this optimization problem.

As said in Chapter 5, the problem of optimal test selection can be also considered as a combinatorial optimization problem. However, this solution is often limited to a centralized view of the system. An interesting perspective is to decentralize or distribute this process. The idea is to solve one integer programming problem per subsystem and to **use the techniques of parallel and distributed integer programming for the test selection problem**.

In Chapter 2, we insisted on the fact that a lot of works studied the decentralization or distribution of model-based diagnosis on discrete event systems. This thesis proposes a solution for continuous systems. However, real systems have become so complex that it is often impossible to model them only considering their continuous aspect or their discrete aspect separately. It is now necessary to consider the systems as the whole, respecting their hybrid behavior. Hybrid systems exhibit both discrete and continuous dynamics. They are usually described as a multi-mode system composed of an underlying discrete-event system (DES) representing the mode changes and various underlying continuous dynamics associated with each mode. A major perspective is to develop techniques **mixing distributed and decentralized diagnosis methods for continuous and discrete event systems, to deal with hybrid systems**. The solution is not a simple superposition of both techniques because of the interaction between continuous and discrete dynamics.

Concerning the application of this work, the decentralized and the distributed algorithms have been experienced on two real case studies. The decentralized diagnosis is applied on a ACDS of a satellite LEO while the distributed diagnosis is tested on a reverse osmosis desalination plant. These very different applications show that our approaches are generic and could be applied to many other real systems. Nowadays, the term "smart systems" includes functions of sensing, actua-

tion and control, network communication, and so on. The application domains are growing: environment, automotive systems, internet of things, health care are some examples of domains that are using smart systems. So they are becoming more and more challenging and the use of relevant diagnosis techniques for such distributed and heterogeneous systems is of interest. The **application of our decentralized or distributed diagnosis method on various smart systems** is among our perspectives.

Another major perspective for industry is the introduction of the human in the process, for example with troubleshooting strategies. In the decentralized hierarchy, the idea is to consider that a fault is not isolable at a certain level. A future work could be to guide a troubleshooting procedure to add manual tests that will lead to isolability. This **interleaving of tests selection and troubleshooting procedures** could be investigates as a sort of active diagnosis on request procedure, including human intervention.

The last perspective is to adapt this work to data-based diagnosis approaches, using the idea of shared variables as shared descriptors. We can consider a distributed or decentralized system, defined in the same way as our approach. The motivation is exactly the same as ours: systems are geographically separated, or developed by different industries with different confidential policies, etc. Each subsystem has a local data-based diagnoser but some faults are not isolable at the local level, so that subsystems have to share some of their descriptors. The goal is obviously to minimize the communication or the number of shared descriptors between subsystems. This can lead to a **decentralized or distributed data-based diagnosis approach**. Going even further, we can also imagine the system to be composed by heterogeneous subsystems, some of which have model-based diagnosers and others data-based diagnosers. The problem of sharing information and minimizing communication formulated in this framework opens perspectives to synergically merge results of two fields.

Bibliography

[Ahmed 2013] F. Ahmed. Modified spiegler-kedem model to predict the rejection and flux of nanofiltration processes at high nacl concentrations. University of Ottawa, vol. 3, pp. 59-66, 2013. (Cited in pages 101 and 102.)

[Alatiqi 1989] I. Alatiqi, A. Ghabris and S. Ebrahim. *System identification and control of reverse osmosis desalination.* Desalination Journal, vol. 75, pp. 119-140, 1989. (Cited in page 103.)

[Armengol 2009] J. Armengol, A. Bregón, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido and L. Travé-Massuyès. *Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches.* In Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09, vol. 42(8) pp. 1480-1485, 2009. (Cited in pages 8, 15, 16, and 17.)

[Atallah 2009] M. J. Atallah and M. Blanton. Algorithms and theory of computation handbook, special topics and techniques. Chapman and Hall/CRC, vol. 2, 2009. (Cited in page 68.)

[Bagajewicz 2004] M. Bagajewicz, A. Fuxman and A. Uribe. *Instrumentation network design and upgrade for process monitoring and fault detection.* AIChE Journal, vol. 50(8), pp. 1870-1880, 2004. (Cited in page 64.)

[Bendick 1991] J. Bendick. Artificial satellites: Helpers in space. Millbrook Press, 1991. (Cited in page 86.)

[Blanke 2006] M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki. Diagnosis and fault-tolerant control. Springer-Verlag Berlin Heidelberg, 2006. (Cited in pages 7, 8, 9, 10, 12, 14, and 17.)

[Boem 2011] F. Boem, R. M. Ferrari and T. Parisini. *Distributed Fault Detection and Isolation of Continuous-Time Non-Linear Systems.* European Journal of Control, vol. 17(5-6), pp. 603-620, 2011. (Cited in page 49.)

[Bondy 1976] J. A. Bondy and U. S. R. Murty. Graph theory with applications, volume 290. Macmillan London, 1976. (Cited in page 67.)

[Bregon 2014] A. Bregon, M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos and B. Pulido. *An event-based distributed diagnosis framework using structural model decomposition.* Journal of Artificial Intelligence Research, vol. 210, pp. 1-35, 2014. (Cited in pages 8 and 28.)

[Butt 2012] S. Siddique Butt, R. Prabel and H. Aschemann. *Fault detection and isolation of a hybrid synchronous machine using parity equations.* Methods

and Models in Automation and Robotics (MMAR) 2012 17th International Conference on, pp. 178-183, 2012. (Cited in page 8.)

[Cassar 1997] J Cassar and M Staroswiecki. *A structural approach for the design of failure detection and identification systems.* In IFAC Conference on Control of Industrial Systems, vol. 30(6), pp. 841-846, 1997. (Cited in page 10.)

[Chanthery 2016] E. Chanthery, L. Travé-Massuyès and S. Indra. *Fault Isolation on Request Based on Decentralized Residual Generation.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46(5), pp. 598-610, 2016. (Cited in pages 28, 35, 40, and 86.)

[Chiang 2012] L.H. Chiang, E.L. Russell and R.D. Braatz. Fault detection and diagnosis in industrial systems. Springer Science Business Media, 2012. (Cited in page 8.)

[Chittaro 2004] L. Chittaro and R. Ranon. *Hierarchical model-based diagnosis based on structural abstraction.* Artificial Intelligence, vol. 155(1-2), pp. 147-182, 2004. (Cited in page 8.)

[Cocquempot 1998] V. Cocquempot, R. Izadi-Zamanabadi, M. Staroswiecki and M. Blanke. *Residual generation for the ship benchmark using structural approach.* In UKACC International Conference on Control (CONTROL 98), vol. 2, pp. 1480 - 1485, 1998. (Cited in page 15.)

[Commault 2008] C. Commault, J-M Dion and S Y Agha. *Structural analysis for the sensor location problem in fault detection and isolation.* Automatica, vol. 44(8), pp. 2074-2080, 2008. (Cited in page 66.)

[Console 2007] L. Console, C. Picardi and D. Theseider. *Framework for Decentralized Qualitative Model-Based Diagnosis.* In 20th International Joint Conference on Artificial Intelligence, pp. 286-291, 2007. (Cited in page 40.)

[Cordier 2007] M. O. Cordier and A. Grastien. *Exploiting independence in a decentralised and incremental approach of diagnosis.* In 20th International Joint Conference on Artificial Intelligence, pp. 292-297, 2007. (Cited in pages 26 and 27.)

[Corfmat 1976] J. P. Corfmat and A.S. Morse. *Decentralized control of linear multivariable systems.* In Automatica, vol. 12(5), pp. 479-495, 1976. (Cited in page 39.)

[Debouk 2000] R. Debouk, S. Lafortune and D. Teneketzis. *Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems.* Discrete Event Dynamic Systems, vol. 10(1), pp. 33-86, 2000. (Cited in page 26.)

[Dessouky 2002] Dessouky and Ettouney. Fundamentals of water desalination. Elsevier, 2002. (Cited in page 98.)

[Dick 1993] J. Dick and A. Faivre. *Automating the Generation and Sequencing of Test Cases from Model-Based Specifications.* In FME'93: Industrial-Strength Formal Methods: First International Symposium of Formal Methods Europe, Odense, Denmark, vol. 670, pp. 268-284, 1993. (Cited in page 65.)

[Ding 2008] S. X. Ding. Model-based fault diagnosis techniques. Springer-Verlag Berlin Heidelberg, 2008. (Cited in page 8.)

[Doran 1966] J. E. Doran and D. Michie. *Experiments with the graph traverser program.* Proceedings of the Royal Society A, vol. 294, pp. 235-259, 1966. (Cited in page 68.)

[Dulmage 1958] A. L. Dulmage and N. S. Mendelsohn. *Coverings of bipartite graphs.* Canadian Journal of Mathematics, vol. 10, pp. 517-534, 1958. (Cited in page 12.)

[Düstegör 2006] D. Düstegör, E. Frisk, V. Cocquempot, M. Krysander and M. Staroswiecki. *Structural analysis of fault isolability in the DAMADICS benchmark.* Control Engineering Practice, vol. 14(6), pp. 597-608, 2006. (Cited in page 10.)

[Edwards 2000] C. Edwards, S. K. Spurgeon and R. J. Patton. *Sliding mode observers for fault detection and isolation.* Automatica, vol. 36(4), pp. 541-553, 2000. (Cited in page 8.)

[Faure 1999] P.P. Faure, L. Trave-Massuyes and H. Poulard. *An interval model-based approach for optimal diagnosis tree generation.* In Proc. DX-99, 10th International Workshop on Principles of Diagnosis, Loch Awe, Scotland, pp. 78-89, 1999. (Cited in page 66.)

[Ferdowsi 2012] H. Ferdowsi, D. Raja and S. Jagannathan. *A Decentralized Fault Detection and Prediction Scheme for Nonlinear Interconnected Continuous-time Systems.* WCCI 2012 IEEE World Congress on Computational Intelligence, pp. 1-7, 2012. (Cited in page 27.)

[Gambier 2009] A. Gambier, T. Miksch and E. Badreddin. *A reverse osmosis Laboratory Plant for Experimenting with Fault-Tolerant Control.* American Control Conference Hyatt Regency Riverfront, St. Louis, MO, USA. Automatica. vol. 27, pp. 3775-3780, 2009. (Cited in pages 98, 99, and 101.)

[Garcia 2011] D. Garcia. *Monitoring and Fault Detection in a Reverse Osmosis Plant using Principal Component Analysis.* IEEE Conference on Decision and Control and European Control Conference, pp. 3044-3049, 2011. (Cited in page 98.)

[Gelso 2008] E. R. Gelso, S. M. Castillo and J. Armengol. *An algorithm based on structural analysis for model-based fault diagnosis.* In Artificial Intelligence

Research and Development. Frontiers in Artificial Intelligence and Applications, vol. 184, pp. 138-147, 2008. (Cited in page 17.)

[Gertler 1991] J. Gertler. *Analytical redundancy methods in fault detection and isolation.* In Preprints of IFAC/IMACS Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS91, pp. 9-21, 1991. (Cited in page 8.)

[Gertler 1998] J. Gertler. Fault detection and diagnosis in engineering systems. CRC Press, 1998. (Cited in page 24.)

[Gertler 2000] J. Gertler. Structured parity equations in fault detection and isolation. Springer London, 2000. (Cited in page 8.)

[Gonzalez-Sanchez 2011a] A. Gonzalez-Sanchez, R. Abreu, H.G. Gross and A. Van Gemund. *RAPTOR: Greedy Diagnostic Prioritization by Ambiguity Group Reduction.* In Proceedings of the 22nd International Workshop on Principles of Diagnosis, DX2011, pp. 84-91, 2011. (Cited in page 66.)

[Gonzalez-Sanchez 2011b] A. Gonzalez-Sanchez, E. Piel, R. Abreu, H. G. Gross and A. J. C. van Gemund. *Prioritizing Tests for Software Fault Localization.* Software: Practice and Experience, vol. 41(10), pp. 1105-1129, 2011. (Cited in page 66.)

[Grbovic 2012] M. Grbovic. *Data Mining Algorithms for Decentralized Fault Detection and Diagnosis in Industrial Systems.* PhD thesis, Temple University Graduate Board, 2012. (Cited in page 50.)

[Hart 1968] P. E. Hart, N. J. Nilsson and B. Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics SSC4, vol. 4 (2), pp. 100-107, 1968. (Cited in pages 68 and 69.)

[Hearn 2005] R. A. Hearn and E. D. Demaine. *PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation.* Theorical Computer Science, vol 343(1-2), pp. 72-96, 2005. (Cited in page 68.)

[Isermann 2006] R. Isermann. Fault-diagnosis systems. Springer-Verlag Berlin Heidelberg, 2006. (Cited in page 8.)

[Isermann 2011] R. Isermann. Fault-diagnosis applications. Springer Heidelberg Dordrecht London New York, 2011. (Cited in pages 7, 8, and 23.)

[Jiang 2014] A. Jiang, Q. Ding and J. Wang. *Mathematical modeling and simulation of SWRO Process based on simultaneous method.* Journal of Applied Mathematics, vol. 2014, pp. 1-11, 2014. (Cited in pages 101 and 103.)

[Khalaf 2008] T. Khalaf. *Estimation of concentration polarization using the Combined Film Theory/Spiegler Kedem Model and Empirical Correlation.* In The 1st Regional Conference of Eng. Sci. NUCEJ Spatial ISSUE vol. 11(2), pp. 322-328, 2008. (Cited in page 103.)

[Khorasgani 2015] H. Khorasgani, D. Jung and G. Biswas. *Structural Approach for Distributed Fault Detection and Isolation.* In 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2015, vol. 48(21), pp. 72-77, 2015. (Cited in pages xi, 12, 28, 43, 50, 54, and 71.)

[Kleer 1987] J. De Kleer and B. C. Williams. *Diagnosing multiple faults.* Artificial Intelligence, vol. 32, pp. 97-130, no. 1, 1987. (Cited in page 66.)

[Korbicz 2012] J. Korbicz, J. M. Kościelny, Z. Kowalczuk and W. Cholewa, editors. Fault diagnosis: Models, artificial intelligence, applications. Springer, 2012. (Cited in page 23.)

[Korf 2010] R. E. Korf. Algorithms and theory of computation handbook. Chapman & Hall/CRC, 2010. (Cited in pages 67, 68, and 69.)

[Krysander 2008a] M. Krysander, J. Aslund and M. Nyberg. *An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis.* IEEE Trans. Syst. Man Cy. A., vol. 38(1), pp. 197-206, 2008. (Cited in pages 16 and 17.)

[Krysander 2008b] M. Krysander and E. Frisk. *Sensor Placement for Fault Diagnosis.* IEEE Trans. Syst. Man Cy. A., vol. 38(6), pp. 1398-1410, 2008. (Cited in page 67.)

[Krysander 2010] M. Krysander, J. Aslund and E. Frisk. *A Structural Algorithm for Finding Testable Sub-models and Multiple Fault Isolability Analysis.* In 21st International Workshop on the Principles of Diagnosis, 2010. (Cited in pages 10, 15, 16, 19, and 20.)

[Leal 2015] R. Leal, J. Aguilar, L. Travé-Massuyès, E. Camargo and A. Ríos-Bolivar. *An Approach for Diagnosability Analysis and Sensor Placement for Continuous Processes Based on Evolutionary Algorithms and Analytical Redundancy.* Applied Mathematical Sciences, vol. 9(43), pp. 2125-2146, 2015. (Cited in page 67.)

[Li 2007] Z. Li, M. harman and R.M. Hierons. *Search algorithms for regression test case prioritization.* IEEE Transactions on software engineering, vol. 33(4), pp. 225-237, 2007. (Cited in page 66.)

[McFall 2007] Ch. McFall. *Fault-Tolerant Control of a reverse osmosis desalination process.* In 8th International IFAC Symposium on Dynamics and Control of Process Systems, Mexico, pp. 161-166, 2007. (Cited in page 98.)

[Niemann 2003] T. Lorentzen M. Blanke H. Niemann. *Structural analysis-A case study of the Romer satellite.* Proc. Fault Detect. Supervision Safety Tech. Processes Symp. (SAFEPROCESS) vol. 36(5), pp. 185-190, 2003. (Cited in page 88.)

[Nilsson 1998] N. J. Nilsson. Artificial intelligence, a new synthesis. Morgan Kaufmann, 1998. (Cited in page 70.)

[Olive 2003] X. Olive, L. Travé-Massuyes and J. Thomas. *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, pp. 615-620, 2003. (Cited in page 66.)

[Palacin 2011] Palacin. New dynamic of reverse osmosis plants with fault simulation. 127-132. Department of Systems Engineering and Automatic Control University of Valladolid, Spain., 2011. (Cited in page 98.)

[Pattipati 1988] K. R. Pattipati and M. G. Alexandridis. *Application of heuristic search and information theory to sequential fault diagnosis.* In Proceedings IEEE International Symposium on Intelligent Control, vol. 4, pp. 291-296, 1988. (Cited in page 65.)

[Pattipati 1992] K.R. Pattipati and M. Dontamsetty. *On a generalized test sequencing problem.* IEEE Trans. Syst. Man Cy. A. vol. 22(2), pp. 392-396, 1992. (Cited in page 65.)

[Patton 2000] R. J. Patton, P. M. Frank and R. N. Clark, editors. Issues of fault diagnosis for dynamic systems. Springer Verlag London, 2000. (Cited in pages 10 and 23.)

[Pencolé 2005] Y. Pencolé and M.-O. Cordier. *A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks.* Artificial Intelligence, vol. 164(1-2) pp. 121-170, 2005. (Cited in page 26.)

[Pérez 2015] C. G. Pérez, L. Travé-Massuyès, E. Chanthery and J. Sotomayor. *Decentralized diagnosis in a spacecraft attitude determination and control system.* Journal of Physics: Conf Series, vol. 659(1) pp. 1-12, 2015. (Cited in page 29.)

[Pérez 2016] C. G. Pérez, Travé-Massuyès E. Chanthery and J. Sotomayor. *Fault Driven Minimal Structurally Overdetermined Set in a Distributed Context.* 27th International Workshop on Principles of Diagnosis DX2016, hal-01392572, 2016. (Cited in pages 50 and 52.)

[Pirmoradi 2009] F. Pirmoradi, F. Sassani and C. de Silva. *Fault detection and diagnosis in a spacecraft attitude determination system.* Acta Astronaut, vol. 65 pp. 710-729, 2009. (Cited in pages 87, 88, and 89.)

[Raghavan 1999] V. Raghavan, M. Shakeri and K. Pattipati. *Optimal and near-optimal test sequencing algorithms with realistic test models.* IEEE Trans. Syst. Man Cy. A., vol. 29(1), pp. 11-26, 1999. (Cited in page 66.)

[Ressencourt 2006] H. Ressencourt, L. Travé-Massuyès and J. Thomas. *Hierarchical modelling and diagnosis for embedded systems.* In 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'2006) pp. 553-558, 2006. (Cited in page 66.)

[Rivas-Perez 2017] R. Rivas-Perez, J. Sotomayor-Moriano and C.G. Perez-Zuniga. *Adaptive Expert Generalized Predictive Multivariable Control of Seawater RO Desalination Plant for a Mineral Processing Facility.* In IFAC World Congress, 2017. (Cited in page 104.)

[Rosich 2007] A. Rosich, R. Sarrate, V. Puig and T. Escobet. *Efficient optimal sensor placement for model-based FDI using an incremental algorithm.* Proc. 46th IEEE Conference on Decision and Control, pp. 2590-2595, 2007. (Cited in page 66.)

[Rosich 2009a] A. Rosich, R. Sarrate and F. Nejjari. *Fuel cell system diagnosis based on a causal structural model.* 7th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS, pp. 534-539, 2009. (Cited in page 17.)

[Rosich 2009b] A. Rosich, R. Sarrate and F. Nejjari. *Optimal sensor placement for FDI using binary integer linear programming.* 20 th International Workshop on Principles of Diagnosis, 2009. (Cited in page 64.)

[Rycroft 2002] M. J. Rycroft. Beyond the international space station: The future of human spaceflight. Springer, 2002. (Cited in page 86.)

[Sarrate 2007] R. Sarrate, V. Puig, T. Escobet and A. Rosich. *Optimal sensor placement for model-based fault detection and isolation.* Proc. 46th IEEE Conference on Decision and Control, pp. 2584-2589, 2007. (Cited in page 64.)

[Sarrate 2012] R. Sarrate, F. Nejjari and A. Rosich. *Sensor Placement for Fault Diagnosis Performance Maximization in Distribution Networks.* In 20th Mediterranean Conference on Control & Automation (MED), pp. 110-115, 2012. (Cited in page 67.)

[Sauter 2006] D. Sauter, T. Boukhobza and F. Hamelin. *Decentralized and autonomous design for FDI/FTC of networked control systems.* IFAC Proceedings Volumes, vol. 39(13) pp. 138-143, 2006. (Cited in page 27.)

[Schmude 2012] R. Schmude. Artificial satellites and how to observe them. Springer, 2012. (Cited in pages 85 and 86.)

[Senthilmurugan 2005] S. Senthilmurugan, A. Ahluwalia and S. Gupta. Modeling of a spiral wound module and estimation of model parameters using numerical techniques. Desalination, 2005. (Cited in pages 101 and 102.)

[Siljak 2011] D. D. Siljak. Decentralized control of complex systems. Dover Publications, 2011. (Cited in page 39.)

[Silver 2005] D. Silver. *Cooperative pathfinding.* AIIDE, 117-122. AAAI Press., 2005. (Cited in page 68.)

[Skovmose 2006] C. Skovmose, V. Cocquempot and R. Izadi-Zamanabadi. *Model Based Fault Detection in a Centrifugal Pump Application.* IEEE Transactions on Control Systems Technology, vol. 14(2) pp. 204-215, 2006. (Cited in page 10.)

[Standley 2012] T. Standley. *Independence Detection for Multi-Agent Pathfinding Problems.* Technical Report, AAAI Technical Report WS-12-10, 2012. (Cited in page 68.)

[Su 2005] R. Su and W. M. Wonham. *Global and local consistencies in distributed fault diagnosis for discrete-event systems.* IEEE Transactions on Automatic Control, vol. 50(12), pp. 1923-1935, 2005. (Cited in page 28.)

[Travé-Massuyès 2006] L. Travé-Massuyès, T. Escobet and X. Olive. *Diagnosability Analysis Based on Component-Supported Analytical Redundancy Relations.* IEEE Trans. Syst., Man, Cybern. Part A: Sys and Humans, vol. 36(6), pp. 1146-1160, 2006. (Cited in pages 10, 15, 17, and 66.)

[Travé-massuyès 2013] L. Travé-massuyès, H. Ressencourt, H. Poulard and J. Thomas. *Method for diagnosing a malfunction of a mechatronic system.* `http://www.freepatentsonline.com/EP2339318B1.html`, 2013. (Cited in page 66.)

[Voutchkov 2016] N. Voutchkov. *Desalination - Past, Present and Future*, 2016. (Cited in page 98.)

[Wang 2007] Y. Wang, T.S. Yoo and S. Lafortune. *Diagnosis of discrete event systems using Decentralized architectures.* Discrete Event Dynamic Systems, vol. 17(2), pp. 233-263, 2007. (Cited in pages 26 and 27.)

[Wang 2017] H. Wang, W. Liu, J. Qiu and P. X. Liu. *Adaptive Fuzzy Decentralized Control for A Class of Strong Interconnected Nonlinear Systems with Unmodeled Dynamics.* IEEE Transactions on Fuzzy Systems, vol. PP(99), pp. 1-1, 2017. (Cited in page 39.)

[Yassine 2008] A. A. Yassine, S. Ploix and J. M. Flaus. *A method for sensor placement taking into account diagnosability criteria.* Int Journal Appl. Math Comput. Sci., vol. 18(4), pp. 497-512, 2008. (Cited in page 67.)

[Zeng 2009] W. Zeng and R. L. Church. *Finding shortest paths on real road networks: the case for A*.* International Journal of Geographical Information Science, vol. 23(4), pp. 531-543, 2009. (Cited in page 69.)

[Zhang 2010] X. Zhang. *Decentralized Fault Detection for a Class of Large-Scale Nonlinear Uncertain Systems.* In American Control Conference, pp. 5650-5655, 2010. (Cited in page 27.)

[Zuliana 2010] I. Zuliana and V. Renuganth. *A study of reaction wheel configurations for a 3-axis satellite attitude control.* Adv. Space Res. vol. 45(6), pp. 750-759, 2010. (Cited in page 89.)

Abstract:

This thesis focuses on fault detection and isolation. Among the different methods to generate diagnosis tests by taking advantage of analytical redundancy, this thesis adopts the approach based on analytical redundancy relations (ARRs). Given a model of the system in the form of a set of differential equations, ARRs are relations that are obtained from the model by eliminating non measured variables. This can be performed in an analytical framework using elimination theory. Another way of doing this is to use structural analysis. Structural analysis is based on a structural abstraction of the model that only retains a representation of which variables are involved in which equations. Despite the rusticity of the abstract model, structural analysis provides a set of powerful tools, relying on graph theory, to analyze and infer information about the system. Interestingly, it applies indifferently to linear or nonlinear systems. This thesis proposes efficient algorithms based on structural analysis for the diagnosis of decentralized and distributed continuous systems as well as for the choice of an optimal set of tests. These algorithms were tested on two industrial case studies.

Keywords : Model based fault diagnosis, Structural analysis, Decentralized and distributed architectures, Planning algorithms, A*.

Auteur: Carlos Gustavo Pérez Zuñiga.
Titre: Analyse Structurelle pour le Diagnostic des Systèmes Distribués.
Directeurs de These: Louise Travé-Massuyès, Elodie Chanthery and Javier Sotomayor.
Date de Soutenance: 21/08/2017.

Résumé:

Cette thèse porte sur la détection et l'isolation de fautes. Parmi les différentes méthodes pour générer des tests de diagnostic utilisant la redondance analytique, cette thèse adopte l'approche par relations de redondance analytique (RRA). Étant donné un modèle du système sous la forme d'un ensemble d'équations différentielles, les RRA sont des relations obtenues à partir du modèle en éliminant les variables non mesurées. Ceci peut être effectué dans un cadre analytique en utilisant la théorie de l'élimination. Une autre solution consiste à utiliser l'analyse structurelle. L'analyse structurelle est basée sur une abstraction du modèle qui ne conserve que les liens entre variables et équations. Malgré son apparente simplicité, l'analyse structurelle fournit un ensemble d'outils puissants, s'appuyant sur la théorie des graphes, pour analyser et inférer des informations sur le système. Par ailleurs, elle a l'avantage de s'appliquer indifféremment sur les systèmes linéaires ou non linéaires. Cette thèse propose des algorithmes efficaces basés sur l'analyse structurelle pour le diagnostic des systèmes continus decentralisés et distribués ainsi que pour le choix d'un ensemble de tests optimal. Ces algorithmes on été testés sur deux cas d'étude industriels.

Mots clés : Diagnostic à base de modèles, Analyse structurelle, Architectures décentralisées et distribuées, Algorithmes de planification, A*.

Discipline: Automatique.

Intitule et Adresse du Laboratoire: Laboratoire d'Analyse et d'Architecture des Systèmes LAAS-CNRS.
7 Avenue du Colonel Roche, 31400 Toulouse, France.