Fault diagnosis and prognosis of hybrid systems using bond graph models and computational intelligence

Yu, Ming

2012

Yu, M. (2012). Fault diagnosis and prognosis of hybrid systems using bond graph models and computational intelligence. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50480>

<https://doi.org/10.32657/10356/50480>

**Fault Diagnosis and Prognosis of Hybrid
Systems using Bond Graph Models
and Computational Intelligence**

Yu Ming

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

*To my dear parents and wife,
for their encouragement and love.*

Acknowledgements

In the completion of my research and this dissertation, I would like to express my sincere gratitude to my supervisor, Professor Wang Dan Wei, for his constant encouragement, support and guidance throughout the study. Without his help, this work would not be possible. His discussions on every issue were always helpful and put forward the way to the solutions.

Also, I would like to thank members of our research group, Dr. Shai Arogeti, Dr. Low Chang Boon, Mr. Huang Lei, and Mr. Ma Shiao as well as Dr. Luo Ming, Ms Zhang Dan Hong, and Ms Zhou Jun Hong from Singapore Institute of Manufacturing Technology for all your discussions and suggestions. My gratitude also goes to the technical staffs at Intelligent Robotics Lab and Autonomous Robotics Research Lab for their help and support during the vehicle experiment process. I am grateful to my friends who made my stay at Singapore a very pleasant one.

Finally, I would like to thank my adorable wife, Ms Xu Juan, for all her love and support. I am as ever, especially indebted to my parents and elder brother for all of their never-ending support they have provided throughout my life.

Table of Contents

Acknowledgements	i
Table of Contents	ii
Summary	vi
List of Figures	ix
List of Tables	xiv
List of Abbreviations	xvi
1 Introduction.....	1
1.1 Background.....	1
1.2 Motivation.....	2
1.3 Literature Review.....	4
1.3.1 Related Works on Fault Diagnosis.....	4
1.3.2 Related Works on Failure Prognosis.....	16
1.4 Problem Formulation	22
1.5 Contributions.....	23
1.6 Organization of the Dissertation	25
2 Preliminaries	27
2.1 Bond Graph.....	27
2.1.1 Bonds, Power and Causality	27
2.1.2 Basic Bond Graph Elements	30

2.1.2.1	Basic 1-Port Elements	30
2.1.2.2	Basic 2-Port Elements	33
2.1.2.3	Junctions	35
2.1.2.4	Sensors	38
2.1.3	Causality Consideration for Basic Bond Graph Elements	38
2.1.3.1	Causality for Basic 1-Port Elements	39
2.1.3.2	Causality for Basic 2-Port Elements	40
2.1.3.3	Causality for Junctions	41
2.1.4	Sequential Causality Assignment Procedure	42
2.2	Hybrid Bond Graph	45
2.2.1	Causality Properties and Causality Assignment for HBG	47
2.1.2.1	Causality Properties of HBG	47
2.1.2.2	Causality Assignment for HBG	52
2.2.2	Global Analytical Redundancy Relations	53
2.2.3	An Illustrative Example	58
2.3	Computational Intelligence	61
2.3.1	Particle Swarm Optimization	61
2.3.2	Differential Evolution	65
2.3.3	Fuzzy Logic System	70
3	Simultaneous Fault Parameters and Mode Change Identification	74
3.1	Quantitative Hybrid Bond Graph Diagnosis	74
3.2	Parameterization of Mode Changes	78
3.3	Identification of Fault Parameters and Mode Switching	80

3.3.1	Comparison Between Gradient and Gradient Free Methods	80
3.3.2	Fault Parameter and Mode Change Estimation.....	81
3.4	Case Study of an Electro-hydraulic Suspension	86
3.5	Case Study of an Electric Circuit.....	105
3.6	Conclusions.....	110
4	Multiple Fault Diagnosis without Fault Pattern and Mode Change	
	Information.....	111
4.1	Types of Fault	112
4.2	Augmented GARRs for Fault Diagnosis	113
4.3	Multiple Fault Isolation without Fault Pattern and Mode Change	
	Information	114
4.4	Fault Estimation via Differential Evolution.....	116
4.5	Illustrative Example	118
	4.5.1 System Description	118
	4.5.2 Simulation Results	122
	4.5.3 Experiment Results	125
4.6	Conclusions.....	129
5	Prognosis of Hybrid Systems with Multiple Incipient Faults	130
5.1	Fundamentals of Prognosis	130
5.2	Multiple Failure Prognosis by Particle Swarm Optimization	132
5.3	An Illustrative Example	139
	5.3.1 System Description	139
	5.3.2 Simulation Results	142
	5.3.3 Experiment Results	146

5.3.4 Comparison Study.....	148
5.4 Conclusions.....	149
6 Online Fault Detection and Isolation in A Mobile Robot Test Bed.....	150
6.1 The CyCab Mobile Robot Test Bed.....	150
6.2 HBG Model of CyCab Electro-hydraulic Steering System	152
6.3 FDI Approach for the Front Steering System	155
6.4 Experiment Study.....	158
6.4.1 Experimental Hardware and Software	158
6.4.2 Experiment Results	162
6.5 Conclusions.....	165
7 Conclusions and Future Research.....	166
7.1 Conclusions.....	166
7.2 Future Research	168
Author's Publications	170
Bibliography	172

Summary

Fault diagnosis and prognosis have received a lot of attention from the Prognostics and Health Management (PHM) society in the past decades. PHM society is a non-profit organization dedicated to the advancement of PHM as an engineering discipline. The society was incorporated in early 2009 as a New York corporation. The flagship event of the society is the Annual Conference of the PHM Society.

As the complexity of industrial systems increases, fault diagnosis become more and more important since it is a crucial means to maintain system safety and reliability. Faults need to be detected close to their occurrence time, so that corrective actions can be taken in a timely manner, and thus avoid catastrophic consequences. These actions include resetting control parameters to compensate for the faults, or reconfiguring the system to minimize the effects of the fault.

Many manmade machines can be modeled as hybrid systems. These systems consist of continuous behavior and discrete states represented by modes. In each mode, the system is governed by continuous dynamics, and different modes correspond to different continuous models. For example, an automobile can operate in multiple modes, such as the acceleration phase and the cruising phase. For each mode of operation, the system dynamics is governed by a different continuous behavioral model. Control signals, such as gear shift, may transition the system from the current mode to a different operating mode. Hybrid system monitoring requires estimation of continuous state variables as well as tracking discrete states (i.e., system modes), and more difficulties are involved due to the mode-varying nature of hybrid system. Subsequently, a framework and effective techniques are required for sophisticated analysis, reliable design, and efficient implementation of hybrid system diagnosis and prognosis algorithms.

This thesis addresses the development and implementation of model based diagnosis and prognosis for hybrid systems using Bond Graph (BG) approach and

computational intelligence. The main contributions of the thesis are: (a) the concept of Augmented Global Analytical Redundancy Relations (AGARRs) which lays the base for describing the fault in hybrid systems of parametric and nonparametric nature; (b) parameterization of mode changes; (c) Fault Detection and Isolation (FDI) and fault identification without fault pattern (i.e., abrupt fault or incipient fault) information; and (d) prognosis of hybrid systems with multiple incipient faults.

The first contribution of this thesis is the introduction of AGARRs to describe the fault in hybrid systems with parametric and nonparametric nature. In actual industrial systems, such as mechatronic systems, faults can occur both on the mechanical part and the electrical part of the plant. Examples like robots, a fault may occur in an actuator, in a sensor, or in a component. A component fault is often considered as a parametric fault, for example, a resistor fault in an electric circuit usually changes the parameter representing resistor value from its nominal value to a faulty value. On the other hand, the fault in sensor or actuator can be described as a nonparametric fault because there is no physical parameter, such as mass or volume, to describe it. The concept of AGARRs concerns faults of nonparametric nature using efficiency factors. The efficiency factor is defined to qualify the severity of the nonparametric fault and thus facilitate the fault identification and failure prognosis.

A technique to parameterize mode changes is developed, in which mode changes are formulated as a function of time, switching time stamps and initial mode coefficients. With the help of the unified formula for mode changes, one can simultaneously identify fault parameters in fault candidates and switching time stamps to evaluate the severity of the faults without mode information. In addition, the unified formula for mode changes also enables failure prognosis without mode information.

Fault diagnosis under different unknown fault patterns is proposed. Since in FDI systems, the nature of the fault, i.e., abrupt or incipient, is unknown in advance, and the fault could be sudden break of belt or slowly blockage of pipe. For sudden change of component parameter, it is often depicted as abrupt fault; whereas the slowly deteriorating of component is usually modeled as incipient fault. It is difficult to detect the incipient fault due to slowly developing nature of the fault and the compensation effect of feedback control. The situation is very harsh when the fault pattern information is unavailable, because it is hard to do fault identification which is

based on the nature of the fault. In order to overcome this problem, the fault pattern selection is introduced as a part of fault identification task.

Prognosis of hybrid systems with multiple incipient faults is developed. Some faults may exhibit slow developing nature due to the wear and ageing of system components. In order to carry out effective maintenance to avoid critical damage to the system, prediction of Remaining Useful Life (RUL) of these faulty components is required. Some dynamic models are employed to describe the degradation behaviors of faulty components. The degradation of each incipient fault needs to follow one of these dynamic models and the exact degradation behavior is unknown beforehand. These incipient faults can happen at a non-detectable mode. After these faults are detected at a detectable mode, an efficient algorithm is proposed to choose the correct degradation behaviors and predict the RUL for those faulty components.

The ultimate goal of the fault diagnosis algorithms is to solve real problems. Diagnosis approaches should be tested and evaluated on real world systems to exhibit practical significance. The Quantitative Hybrid Bond Graph (QHBG) based FDI framework is studied in details using a complex mechatronic system – a CyCab mobile robot test-bed.

List of Figures

1.1	Residual generators.	7
2.1	Engineering multiport connection	28
2.2	Word bond graph for system of Figure 2.1.	28
2.3	A simple bond graph representation	30
2.4	Representation of 1-port passive elements.	30
2.5	1-port resistor in electrical domain	31
2.6	1-port capacitor in mechanical domain	31
2.7	1-port inertia in mechanical domain.	32
2.8	1-port active elements in bond graph.	33
2.9	Bond graph representation of a transformer element.	34
2.10	A transformer element in mechanical domain.	34
2.11	Bond graph representation of a gyrator element.	35
2.12	An electrical gyrator element.	35
2.13	Bond graph representations of 1-junction and 0-junction.	36
2.14	An example of bond graph representation involving 1-junction.	37
2.15	An example of bond graph representation involving 0-junction.	37
2.16	Bond graph representations of sensors.	38
2.17	Bond graph causality assignment of sources.	39
2.18	Bond graph causality assignment of resistor.	39
2.19	Bond graph causality assignment of capacitor.	40

2.20 Bond graph causality assignment of inertia.	40
2.21 Bond graph causality assignment of transformer.	40
2.22 Bond graph causality assignment of gyrator.	41
2.23 Bond graph causality assignment of 1-junction.	41
2.24 Bond graph causality assignment of 0-junction.	42
2.25 An example of a BG with causality.	43
2.26 A RLC electric circuit.	44
2.27 Causality assignment process of the electric circuit.	44
2.28 Diagram of hybrid systems.	45
2.29 A simple hybrid system: electrical circuit.	46
2.30 HBG model of the circuit when the switch is ON.	46
2.31 HBG model of the circuit when the switch is OFF.	46
2.32 A hybrid system of electrical circuit.	48
2.33 HBG of with a causality assignment when the switch is OFF.	48
2.34 HBG with an alternate causality assignment when the switch is OFF.	48
2.35 1-controlled junction connected with 0-junction.	49
2.36 0-controlled junction connected with 1-junction.	50
2.37 A two-tank system.	58
2.38 The two-tank plant acaused HBG.	59
2.39 The two-tank plant DHBG.	59
2.40 Movement of a particle.	63
2.41 Two dimensional example of an objective function showing its contours and the operation of generating mutant vector $F \cdot (X_{r_1j}^G - X_{r_2j}^G)$ using strategy DE/rand/1/bin.	67

2.42 A pictorial example for the binary crossover in DE ($rnbr(i) = k = 7$)	68
2.43 A pictorial example for the exponential crossover in DE.	69
2.44 Basic configuration of a fuzzy logic system.	71
2.45 Fuzzification process based on triangular membership functions.	72
3.1 Architecture of quantitative diagnosis framework.	75
3.2 Mode change sequence when the initial state is 1.	79
3.3 Mode change sequence when the initial state is 0.	80
3.4 Particle structure of HPSO algorithm	82
3.5 Fuzzy input and output membership functions.	85
3.6 Block diagram of AHPSO algorithm.	86
3.7 Model of a quarter car.	86
3.8 Actuator with four way spool valve.	87
3.9 HBG of quarter-car electro-hydraulic suspension system.	91
3.10 DHBG of quarter-car electro-hydraulic suspension system.	93
3.11 Simulink model representing constitutive relation of R -element.	97
3.12 Simulink model of constitutive relation of 0 -junction.	97
3.13 HBG model of Figure 3.9 in Simulink environment.	98
3.14 Road profile of the first scenario.	100
3.15 Road profile of the second scenario.	100
3.16 Residual responses of the first scenario.	101
3.17 Residual responses of the second scenario.	102
3.18 A hybrid system: an electric circuit.	105
3.19 HBG of the electric circuit	105
3.20 DHBG of the electric circuit.	105

3.21 Experiment setup.	108
3.22 High speed photomos switching relay	108
3.23 Switching signal of the electric circuit.	108
3.24 Residual responses of the electric circuit.	109
4.1 Individual structure of HDE algorithm considering multiple faults.	117
4.2 A model based health monitoring framework for hybrid systems	117
4.3 An electrical hybrid system	118
4.4 DHBG of the circuit.	118
4.5 Fault profiles for simulation.	122
4.6 Switches programming for simulation.	122
4.7 Simulation results for the first scenario	123
4.8 Mean fitness evolution versus iterations for simulation.	124
4.9 Fault profiles for experiment	126
4.10 Switches programming for experiment.	126
4.11 Experiment results for the second scenario.	127
4.12 Mean fitness evolution versus iterations for experiment.	128
5.1 Degradation profile	132
5.2 Particle structure of AHPSO algorithm.	137
5.3 Block diagram of the MAHPSO based prognosis method.	138
5.4 Particle structure of AHPSO algorithm considering multiple faults	138
5.5 A hybrid system: an electric circuit.	139
5.6 DHBG of the electric circuit	139
5.7 Degradation profile in electrical circuit	142
5.8 Switches programming for simulation.	143

5.9	Simulation residual responses for resistance fault and sensor fault.	144
5.10	Mean fitness evolution versus iterations for simulation	145
5.11	An experimental setup for FDI and prognosis.	146
5.12	Degradation profile in β_{v_1}	146
5.13	Switches programming for experiment	146
5.14	Experimental residual responses for actuator fault.	147
5.15	Mean fitness evolution versus iterations for experiment	148
6.1	The steering system of CyCab mobile robot	150
6.2	Piston internal leakage simulated by a valve	151
6.3	DHBG model of the front steering system.	152
6.4	The Ackerman steering mechanism.	154
6.5	Structure and hardware configuration of Cycab steering system	158
6.6	Simulink model of data acquisition for the incremental encoder and absolute encoder signals.	159
6.7	Simulink model of data acquisition for the pressure sensor signals.	160
6.8	Simulink model of FDI process.	160
6.9	Internal structure of Simulink model of FDI process.	161
6.10	Real time residual responses for a fault mode- burnt driver fault in a_1	163
6.11	Real time residual responses for a sensor fault - pressure sensor fault in p	164
6.12	Real time residual responses for a parameteric fault - internal leakage in R_l	164

List of Tables

2.1	Effort and flow in different domains.	29
3.1	Rules tables for AHPSO.	85
3.2	MD-FSM at $a_1 = 1$, $a_3 = 1$	96
3.3	MD-FSM at $a_1 = 0$, $a_3 = 1$	96
3.4	MD-FSM at $a_1 = 1$, $a_3 = 0$	96
3.5	MD-FSM at $a_1 = 0$, $a_3 = 0$	96
3.6	Nominal physical parameters of the electro-hydraulic suspension	99
3.7	Thresholds for the two scenarios.	102
3.8	Parameter sensitivity analysis for identification.	104
3.9	Comparison results using different methods.	105
3.10	MD-FSM at $a = 1$	107
3.11	MD-FSM at $a = 0$	107
3.12	Parameter sensitivity analysis for identification	110
3.13	Comparison results using different methods	110
4.1	MD-FSM at $a = 0$	115
4.2	MD-FSM at $a = 1$	115
4.3	MCSM of the example	115
4.4	The MCSM of the electrical circuit.	120
4.5	The MAT of the electrical circuit.	120

4.6 MD-FSM at $[a_2 \ a_1] = [1 \ 1]$	121
4.7 MD-FSM at $[a_2 \ a_1] = [0 \ 1]$	121
4.8 MD-FSM at $[a_2 \ a_1] = [1 \ 0]$	121
4.9 MD-FSM at $[a_2 \ a_1] = [0 \ 0]$	121
4.10 Summary for simulation results.	125
4.11 Summary for experiment results	128
5.1 MD-FSM at $a = 1$	135
5.2 MD-FSM at $a = 0$	135
5.3 MD-FSM at $a_1 = 0, \ a_2 = 1$	141
5.4 MD-FSM at $a_1 = 1, \ a_2 = 0$	141
5.5 MD-FSM at $a_1 = 0, \ a_2 = 0$	142
5.6 MD-FSM at $a_1 = 1, \ a_2 = 1$	142
5.7 Summary for simulation results	145
5.8 Summary for experiment results	148
5.9 Comparison results using different methods.	149
6.1 MD-FSM at mode $[a_1 \ a_2] = [1 \ 1]$	157
6.2 The MCSM of the steering system	157
6.3 Physical parameters of the CyCab front steering system	162
6.4 Thresholds for the mobile robot FDI.	163

List of Abbreviations

AGA	Adaptive Genetic Algorithm
AGARR	Augmented Global Analytical Redundancy Relation
AHPSO	Adaptive Hybrid Particle Swarm Optimization
AI	Artificial Intelligence
ARR	Analytical Redundancy Relation
BDE	Binary Differential Evolution
BG	Bond Graph
BPSO	Binary Valued Particle Swarm Optimization
CBM	Condition Based Maintenance
DE	Differential Evolution
DHBG	Diagnostic Hybrid Bond Graph
EA	Evolutionary Algorithm
FDI	Fault Detection and Isolation
FSM	Fault Signature Matrix
GA	Genetic Algorithm
GARR	Global Analytical Redundancy Relation
HBG	Hybrid Bond Graph
HDE	Hybrid Differential Evolution
HMM	Hidden Markov Models
HPSO	Hybrid Particle Swarm Optimization
IMC	Initial Mode Coefficients

IMM	Interacting Multiple Model
MAHPSO	Multiple Adaptive Hybrid Particle Swarm Optimization
MAT	Mode AGARRs Table
MCSM	Mode Change Signature Matrix
MD-FSM	Mode Dependent Fault Signature Matrix
MGY	Modulated Gyrator
MHDE	Multiple Hybrid Differential Evolution
MTF	Modulated Transformer
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PHM	Prognostics and Health Management
PSO	Particle Swarm Optimization
QBG	Qualitative Bond Graph
QHBG	Quantitative Hybrid Bond Graph
QSIM	Qualitative Simulation
RDE	Real Differential Evolution
RPSO	Real Valued Particle Swarm Optimization
RUL	Remaining Useful Life
SCAP	Sequential Causality Assignment Procedure
SCAPH	Sequential Causality Assignment Procedure for Hybrid Systems
SOC	State Of Charge
SVC	Support Vector Classifier
SVM	Support Vector Machine
TCG	Temporal Causal Graph

Chapter 1

Introduction

1.1 Background

Safety and reliability are of primary importance for modern complex engineering systems, which in turn calls for sophisticated and efficient techniques to detect, isolate and identify faults. However, no matter how good the system design is, system components deteriorate over time due to certain stress or load in the real environment. System health monitoring is a key feature for early detection of faults, failure prevention, reliability and condition based maintenance. A health monitoring system needs detect a fault or failure in a timely manner so that necessary safety is maintained and the replacement of faulty components can be performed effectively to ensure normal operations [9]. A health monitoring system integrates two main aspects, namely fault diagnosis and failure prognosis. Fault diagnosis and prognosis are complementary tasks since diagnosis is a “static” indicator whereas prognosis is a “dynamic” indicator. The task of diagnosis is to indicate whether or not a fault has occurred, determine the occurrence of a specific fault out of a set of potential fault candidates and also give some information about the severity of the fault. Prognosis deals with fault prediction before failure occurring. The most widely used prognosis is to predict how much time is left before a failure occurs, i.e. Remaining Useful Life (RUL), given the current machine condition and past operation profile [10].

Many manmade systems, such as switched mode power converter and high-speed printers, can be modeled as hybrid systems. These systems consist of continuous dynamics and discrete states represented by modes. In each mode, the system is

governed by continuous dynamics and different modes correspond to different continuous models. A hybrid system has the benefit of encompassing a larger class of systems within its structure, allowing for more flexibility in modelling dynamic phenomena. In the case of a hybrid system health monitoring, and due to its mode varying nature, more difficulties are involved.

Discrete-event approaches for diagnosis in a hybrid system usually rely on the temporal discrete event evolution and abstract away the detailed information supplied by continuous states. These methods cannot detect and isolate the faults exhibiting small continuous variations. Continuous system fault diagnosis and prognosis approaches typically consider system evolution as a function of time. They cannot handle discrete mode changes of hybrid systems. These above-mentioned works cannot apply to hybrid systems because hybrid system monitoring requires tracking of continuous state variables and estimation of discrete states (i.e., system modes). Discrete events force the system to move from one mode to another; these changes are referred to as mode-changes. Some mode changes may take place autonomously as a result of the continuous evolution of system state variables. Some transitions are initiated by a discrete event, such as a supervisory controller (a jump from one mode to another). The main difficulty when designing fault diagnosis of hybrid systems is due to unpredicted mode changes caused by unknown discrete dynamics. Moreover, prognosis for hybrid systems is challenging due to the fact that continuous dynamics and system mode are mutually dependent and intertwined.

1.2 Motivation

Model-based fault diagnosis of hybrid systems is received much interest recently. The performance of model based FDI depends on model accuracy and difficulty increases with system complexity. A model must be constructed at a level of detail where system behaviors can be mapped to system components and parameters. If a proper model for hybrid systems is utilized, model based method can achieve more accurate performance compared with data driven method. Model based method typically utilizes a set of residuals called Analytical Redundancy Relations (ARRs) which represent the discrepancy between information obtained from the actual system and that generated by a model of the system. It is not difficult to carry out fault

diagnosis for hybrid systems provided that the mode information is available before and after the fault occurrence. If mode information is unavailable before a fault happens, it is difficult to realize fault diagnosis because the abnormal behavior of residual may be caused by a fault, or a mode change. On the other hand, when mode is unknown after a fault is detected, it is hard to carry out fault isolation and fault identification. Consequently, a systematic method is required to deal with fault diagnosis without mode information.

Fault diagnosis techniques rely fundamentally on the information about the process obtained by sensors and the control signals by means of the actuators. Consequently, severe problems arise if the input and output signals must be considered to be possibly sources of fault, i.e., the measurements may be corrupted or the physical effects of actuators may distinct from the expected ones. Hence, fault diagnosis of sensors and actuators has received particular attention. In addition, the fault nature may be abrupt or incipient. In most real applications, it is difficult to know the fault nature in advance. For example, for a mobile robot, the fault could be the slowly wearing of oil piston which leads to the cylinder internal leakage, or the suddenly broken belt. In some cases, multiple faults may happen simultaneously in components, sensors or actuators. All these factors add to the complexity of fault diagnosis in hybrid systems. It is desirable to design a method which can handle the above mentioned issues.

Failure prognosis for hybrid systems is a new topic. Prognosis for hybrid systems is challenging due to the fact that continuous dynamics and system mode are mutually dependent and intertwined. Mode information plays a vital role in designing prognosis method for hybrid systems. Traditional failure prognosis of continuous systems tracks the fault degradation and predicts the RUL for single component, while it cannot monitor several fault sources simultaneously. Therefore, prognosis under single fault assumption could provide good insights into Condition-Based Maintenance (CBM) analysis. A real complex system consists of many components, and a method capable of adapting to handle multiple incipient faults can provide several additional advantages but is not available up date.

1.3 Literature Review

This part reviews various strategies and methods on fault diagnosis and prognosis that have been widely used.

1.3.1 Related Works on Fault Diagnosis

The overall concept of fault diagnosis consists in the following three essential tasks:

- 1) Fault detection: indication of the occurrence of faults, which lead to undesired or unacceptable behavior of the system.
- 2) Fault isolation: localization of different faults.
- 3) Fault identification: determination of the type and magnitude of the fault.

Model based diagnostic method

In model based methods, models serve as knowledge representation of a large amount of structural, functional and behavioral information and their relationship. This representation is utilized to create complex cause-effect reasoning leading to construction of powerful and robust automatic diagnosis and isolation systems [17].

Qualitative model based approach is applicable when the numerical model of the system is unavailable. It utilizes qualitative abstractions to model complex systems while model structure is well defined. The models used in qualitative methods are relatively simple compared with numerical models. The sensitivity of fault diagnosis system to modeling errors and sensor noises may be alleviated [2]. Qualitative Simulation (QSIM) is a widely used modeling tool to describe continuous model qualitatively [3]. This approach is intended to simulate the behavior of physical systems using qualitative values, rather than providing explanations for behaviors of physical processes. In [4], several faulty models are built using QSIM. The observed faulty behavior from real system is compared with that from the faulty models to choose the faults set which occur in the system. However, the faulty models determination needs prior knowledge. In [7, 8], QSIM based fault diagnosis is presented to handle multiple faults in continuous devices. The qualitative modeling framework quantizes the state space utilizing landmark values and specifies qualitative relations between the quantized states, which leads to a set of qualitative

differential equations. A fuzzy qualitative simulation method is developed in [11]. The advantage of this method over QSIM based fault diagnosis is that if the observed behavior cannot match the predicted behavior of any faulty model, the candidate generator will check the modified models whose predicted behavior can match the observed one. Therefore, the modified models selected from the generator will determine the fault candidates. The fuzzy qualitative simulation method can provide more precise information than QSIM based method because the utilization of fuzzy sets leads to a more accurate representation of time. In continuous system diagnosis, time is an important factor to be considered during algorithm design. However, it is difficult to choose appropriate fuzzy sets number and membership function which is a common problem for fuzzy logic system design. In addition, there is no efficient method to determine the number of modified models to be searched by candidate generator.

In [12], a Qualitative Bond Graph (QBG) method is proposed. QBG is developed based on the combination of qualitative reason and bond graph modeling theory in order to benefit from both methods. In QBG, qualitative equations form bond graph models instead of the differential equations are used for fault analysis. These equations represent the components physical variables, locations, and their functional relations can be stated directly from the model. This is particularly suitable for model based fault diagnosis because possible faults can be localized through an analysis of the interrelations of the component states and observed abnormal behavior qualitatively. Such qualitative algorithm is done using available measurement, i.e. the history of past data. Therefore, the qualitative behavior equations are always written in differential causality [13].

For passive elements, qualitative behavior equations are defined as

$$\text{R-element: } e(k) = Rf(k),$$

$$\text{C-element: } f(k) = C[e(k) - e(k-1)],$$

$$\text{I-element: } e(k) = I[f(k) - f(k-1)].$$

where k denotes the current time and $k-1$ represents the time at previous sample. Since only qualitative information is considered, the actual sampling time T , used for taking derivatives, does not appear in these equations.

A qualitative space $\{[+1],[+],[0],[-],[-1],[?]\}$ is defined to represent system behaviors and component states for qualitative fault diagnosis [14]. For system measurements, $[+]$ and $[-]$ stand for the positive and negative values in the measurement, respectively. $[0]$ represents the boundary between $[+]$ and $[-]$, while $[?]$ denotes an uncertain value. $[+1]$ and $[-1]$ stand for very large and negative values, respectively. For power variables E and F , $[+1]$, $[-1]$ and $[0]$ represent different abnormal behaviors due to system faults, whereas $[+]$ and $[-]$ denote normal states. For system components R , C and I , $[+]$ represents a component blocked, e.g. valve blocked, which hinders power delivery, and $[0]$ denotes component leakage or short circuit. Since the operations on qualitative variables are different from numerical ones, a set of qualitative operators are defined as $\{+, -, \times, /, =\}$ and these qualitative operators have the same mathematical meanings as their counterparts in numerical domain. The inference mechanism of qualitative equations is different from normal mathematical equations according to the definitions of qualitative operation tables [14]. For example, consider qualitative equation $[+1] = [+1] + [X]$, a set of solutions $[X] = \{[0], [+], [+1]\}$ can be obtained. Since from these qualitative operation tables, $[+1] = [+1] + [0]$, $[+1] = [+1] + [+]$ and $[+1] = [+1] + [+1]$ are correct. While $[+] = [+1] + [-1]$ is not the equation which provides the solution according to the qualitative operation tables. If the equation is solved using a standard mathematical operation, the equation is rewritten as $[X] = [+1] - [+1] = [0]$, only one unique solution is derived. Unlike fault-tree analysis and fuzzy-logic-based fault diagnosis, a priori knowledge about the cause-effect relationships of system faults is not required for QBG based approach [15, 16]. This makes the QBG based diagnosis more robust in order to accommodate different unforeseeable faults. However, due to the imprecise characteristic of qualitative representation, this method has difficulty in detecting and localizing incipient faults [14, 94]. Due to the limited discriminatory ability of qualitative method, it is difficult to deal with the isolation of multiple faults. Moreover, the diagnosis results of QBG based method usually contain a set of suspected faults, which is not enough for some

diagnosis tasks. In order to accurately know the fault information, i.e. fault severity, to achieve fault tolerance performance, numerical models are preferred.

Quantitative fault diagnosis method checks the consistency between actual system and its behavior model. Consistency checking is usually achieved through a comparison between the information obtained from the real system and that from a behavioral model. The resultant differences are called residuals. Each residual should be normally zero or close to zero when the system is normal, but should be distinguishably different from zero when a fault occurs [21]. Fault is detected by monitoring the trend of the residuals, which usually involves setting a fixed threshold on a residual quantity. The models should be insensitive to modeling errors while at the same time sensitive to faults. The major elements of the FDI system are a residual generator and a residual evaluator. Residual generation can be developed using different methods, such as observer, parity relations (also called analytical redundancy relations) and parameter estimation based methods. Observer based fault diagnosis is a well known analytical model based FDI scheme, which compares the actual output from a system with reference output from an analytical model. ARRs based method works by evaluating the constraint using input signals, sensor data and parameter values from the monitored system [22, 87]. The constraint, written in symbolical form, only contains known variables. In other word, ARRs are static or dynamic constraints which link the time evolution of the known variables when a system operates according to its normal operation model. ARRs are generated through a design procedure that eliminates the unobservable variables in the system equations [89]. This design procedure can be complicated for complex systems, especially those with

Figure 1.1: Residual generators

nonlinear behaviors and multiplicative faults. These two approaches of residual generation for FDI in real time are shown in Figure 1.1. When there is a fault in the system, some residuals must deviate sufficiently such that the deviation is distinguishable from the residual response during normal operation.

Dedicated observer approaches belong to observer based method and they can be used for sensor or actuator fault diagnosis [18]. For sensor fault isolation, a number of Luenberger observers are built where each observer is connected to a single output. For actuator fault detection, each observer is connected to a single input. The basic idea is to have a bank of observers where each observer only depends on one fault and is decoupled from all other faults and all disturbances. Hence, this method can detect the same number of faults simultaneously as the number of observers [19]. Parameter estimation approaches use the assumption that the faults of a dynamical system are reflected in the physical parameters such as mass, friction, resistance, etc. Identification of the parameters of the mathematical model can be used for fault detection and isolation [1, 20].

Another element of quantitative based method is residual evaluation, which determines whether any faults have occurred by checking the residuals and their trends. The decision making rules usually are designed specifically for different process [86].

Robust fault diagnosis tries to minimize misdetection and false alarms by considering the residual noises. Misdetection, or miss alarm, means missing to detect the presence of an actually occurred fault. On the contrary, false alarm refers to an indication of fault which in fact does not happen. There are various approaches to generate robust residuals, which are insensitive to modeling uncertainties and measurement noises. One of the robust methods, known as active approach, is based on generating residuals that are insensitive to modeling uncertainties, but sensitive to faults. Some techniques like unknown input observer and robust parity equations are proposed to achieve active robust performance [23-25]. An alternative approach to achieve robustness, called as passive, attempts to accomplish robustness in the decision making stage. In these methods, the effect of the parameter uncertainty is propagated to the residuals and then an adaptive threshold is used to envelop these residuals to achieve robustness [26, 27]. In [27], a robust FDI with respect to parameter uncertainties is proposed using bond graph modeling approach in the linear

fractional transformation configuration. The robust ARR's are generated directly from a bond graph model. The ARR's consist of two perfectly separate parts, a nominal part that describes the system operation, and an uncertain part which is used for residuals' sensitivity analysis and evaluation [88]. The decision procedure for residual evaluation is improved using the detectability indexes, which represent the ability of the system to detect faults in presence of parameter uncertainties. It is concluded that the robustness in fault detection can be achieved in the residual generation or in the decision making stage. The residual preprocessing in active approach is analytical, while post processing in passive approach is usually numerical. Active method is easy to implement but only can deal with limited types of system nonlinearity. Passive technique is not constrained by the forms of system nonlinearity; however, it is less robust compared with passive method due to the fact that there could be some misdetection of small magnitude faults. Some improvement can be made by using cumulative sums to detect small magnitude faults, but likelihood of false alarm increases. Consideration of tradeoff is then made depending on specific system.

Hybrid automata is a commonly used scheme for hybrid system modeling. In a hybrid automaton, hybrid system behavior is defined by the continuous dynamics and discrete mode changes that alter the continuous dynamics. Mode transitions are controlled or autonomous. In hybrid system diagnosis, two different types of faults are defined. The first is a parametric fault, where one or more model parameters are deviating from their nominal value to an unknown value. The second type is fault-mode; in this case, the faulty state is known a priori, and can be modeled by known parameters only. Two examples are short-circuits and open-circuits faults of power switches [33]. Fault modes can be detected and isolated by mode identification techniques. In [28, 29], a model-based mode-identification approach is presented, where a bank of N continuous observers (each represents a suspected system's mode) is used simultaneously, in a hybrid system estimation framework. An observer-based mode identification is also proposed in [30], for a switching linear system. The problem with these techniques is that no parametric fault is considered, and the system's actual behavior must match with one of a predefined set of modes.

A discrete model-based approach for mode estimation and fault detection of hybrid systems is presented in [31, 32], where the authors used a timed Petri net to abstract away the continuous dynamics and to describe only the temporal discrete event

evolution of the system. This method, although general, requires domain specific knowledge for offline generation of a fault-symptom table. Moreover, training data (from reliability studies and testing) is necessary for computation of individual faults prior probability distributions and signal-event templates to characterize mode transitions.

In [36, 37], FDI in a hybrid system is based on structured parity residuals (also known as ARR_s), and the model is a hybrid automaton. Two types of faults are considered: faults that affect the current mode behavior, and faults affecting discrete evolution trajectory. Faults of the second type are detected by a mode tracking technique, based on residuals consistency and discrete dynamical model. If a fault of the first type occurs, mode identification is based on only those residuals which are robust to the fault.

In [34, 35], a qualitative-quantitative diagnosis framework is developed for abrupt parametric faults in hybrid systems. The fault detection is based on a hybrid observer, which integrates a Kalman filter (for continuous tracking) and a mode-change detector. The discrete events are either known controlled mode changes or autonomous mode changes (triggered by continuous state variables). Once the mode-change conditions are detected, a finite state machine is utilized to determine the new mode. The state-space model of the new mode is dynamically computed and applied to the Kalman filter. If a fault is detected, the hybrid observer is unable to track autonomous mode changes. Consequently, fault-hypotheses are generated in all previous modes up the level of system diagnosability. A qualitative approach based on Temporal Causal Graph (TCG) is utilized to narrow the set of fault hypotheses, and only those hypotheses which are consistent with the system's observations are considered for fault parameter identification. In this work, modes are not identified based on their continuous dynamics, but only hypothesized from the mode-change conditions. Furthermore, when data is collected for fault parameter identification, the mode is assumed to be persistent.

A mode identification method for hybrid systems in the presence of fault is proposed in [38]. The method is presented as a module of a Hybrid Bond Graph (HBG) based quantitative health monitoring framework. This framework utilizes a concept of Global Analytical Redundancy Relations (GARR_s), and integrates several steps and modules. Mode identification of hybrid system in the presence of faults is

based on continuous evaluation of all ARR_s given by the ARR-set. The ARR-set is generated from the GARR_s of the hybrid system. Dependent GARR_s increase the number of ARR_s in the ARR-set and improve Mode-Identification. In the online process, ARR_s are grouped in four subsets according to their certainty and consistency. Certain ARR_s represent model constraints of only normal parameters, and uncertain ARR_s are possibly influenced by the fault. Each subset of ARR_s provides different types of information, and the information from all ARR_s is merged together to yield the mode identification result. The mode identification process is based on nine basic rules, and two implementation methods (parallel and iterative) are proposed. Parametric faults are isolated by the global fault signature matrix. The isolated faults are given in the fault candidates set which is utilized for ARR classification, to certain and uncertain. The mode identification process has the potential to refine the fault candidates set, and subsequently to improve mode identification, by letting more ARR_s to be classified as certain.

A Quantitative Hybrid Bond Graph (QHBG) based FDI framework to address the FDI problems for hybrid systems is developed in [39, 40]. The framework utilizes the unified information of the HBG to monitor a hybrid system efficiently and effectively. This framework consists of a GARR alarm generator, a fault detection module, a fault isolation module, a fault estimator, and a mode tracker. These modules are based on the HBG of the hybrid system. The mode tracker is utilized to determine the instantaneous operating mode using sensors and input information obtained from the hybrid system [42, 43]. This instantaneous mode information allows users to evaluate the GARR_s for residuals effectively at all operating modes. With these residuals, one can detect, isolation, and estimate the size of a fault. The fault detection module decides whether or not a fault has occurred when any of the residual signals is non-zero. For those fault isolable parameters, the GARR_s generate a set of unique residuals that allows the faulty parameter to be identified. For those non-isolable but detectable parameters, the module will select a set of potential fault candidates. Finally, the parameter estimation module estimates the selected fault parameters to access the health status of the system [41]. This information can also be used to refine the non-isolable parameter.

Data driven diagnostic method

In contrast to the model based approaches where a priori knowledge (either quantitative or qualitative) about the process is needed, in data driven based methods, only the availability of large amount of historical process data is needed. There are different ways in which this data can be transformed and presented as a priori knowledge to a diagnostic system. This is known as feature extraction. Data driven diagnostic approaches can be broadly classified as statistical methods and artificial intelligence methods.

Cluster analysis, as a multivariate statistical analysis method, is a statistical classification method that groups signals into different fault categories on the basis of the similarity of the characteristics or features they possess. It seeks to minimize within-group variance and maximize between-group variance. The result of cluster analysis is a number of heterogeneous groups with homogeneous contents. There are substantial differences between the groups, but the signals within a single group are similar. A natural way of signal grouping is based on certain distance measures or similarity measure between two signals. These measures are usually derived from certain discriminant functions in statistical pattern recognition. Ding *et al.* [142] introduced a new distance metric termed quotient distance for engine fault diagnosis. Pan *et al.* [143] proposed an extended symmetric Itakura distance for signals in time-frequency representations such as the Wigner-Ville distributions. Other than distance measures, feature vector correlation coefficient is also a similarity measure commonly used for signal classification in machinery fault diagnosis [144].

A technique called Support Vector Machine (SVM) is usually employed to optimize a boundary curve in the sense that the distance of the closest point to the boundary curve is maximised. In [145], a novel frequency pattern and competent criterion are introduced for short-circuit-fault recognition in permanent-magnet synchronous motors. The frequency pattern is extracted from the monitored stator current analytically and the amplitude of sideband components at these frequencies is introduced as a proper criterion to determine the number of short-circuited turns. The occurrence and the number of short-circuited turns are predicted using SVM as a classifier. In view of the bad diagnosing capability of standard Support Vector Classifier (SVC) for fault diagnosis pattern series with Gaussian noises, Gaussian

function is used as loss function of SVC and a new SVC based on Gaussian loss function technique, by name g-SVC, is proposed in [146]. The results of its application to car assembly line diagnosis indicate that the diagnosing method is effective and feasible. A dual features functional SVC approach that uses both first and second derivatives of degradation profiles for early detection of faulty batteries with the reduced error rate is developed in [147]. The modified floating search algorithm for the repeated feature selection with newly added degradation path points is presented to find a few good features for the enhanced detection while reducing the computation time for online implementation. After that, an attribute sampling plan considering time-varying classification errors is presented to determine the optimal number of test cycles and sample sizes by minimizing the proposed cost function. The proposed method can be applied in a wide range of manufacturing processes to assess time-dependent quality characteristics.

Multivariate statistical techniques are powerful tools capable of compressing data and reducing its dimensionality so that essential information is retained and easier to analyze than the original huge data set; and they are also able to handle noise and correlation to extract true information effectively. The main function of multivariate statistical techniques is to transform a number of related process variables to a smaller set of uncorrelated variables. Principal Component Analysis (PCA) is a standard multivariate technique. Theoretically, PCA is based on an orthogonal decomposition of the covariance matrix of the process variables along directions that explain the maximum variation of the data. The main purpose of using PCA is to find factors that have a much lower dimension than the original data set which can properly describe the major trends in the original data set. A two-dimensional dynamic principal component analysis is proposed to model and monitor such two-dimensional dynamic batch processes, in which support region determination is a key step [148]. A proper support region ensures modeling accuracy, monitoring efficiency, and reasonable fault diagnosis. The support region determination method is practicable in many situations but still has certain limitations. To overcome these shortcomings, a two-dimensional dynamic principal component method with an improved support region determination procedure is developed, by considering variable partial correlations and performing iterative stepwise regressions. Such a procedure expands support region batch by batch and is a generalization of the autoregressive model order selection to the two-

dimensional batch process cases. Some unique characteristics of semiconductor manufacturing processes have posed challenges for fault diagnosis applications, such as nonlinearity in most batch processes, and multimodal batch trajectories due to product mix. To explicitly account for these unique characteristics, a pattern recognition based fault detection method utilizing the k-nearest-neighbor rule was developed. In k-nearest-neighbor rule, historical data are used directly as the reference of normal process operation to determine whether a new measurement is a fault. Therefore, for processes with a large number of variables, it can be computation and storage intensive, and may be difficult for online process monitoring. To address this difficulty, a fast pattern recognition based fault detection method, termed principal component-based k-nearest-neighbor is proposed in [149], which takes advantages of both PCA for dimensionality reduction and k-nearest-neighbor rule for nonlinearity and multimode handling.

Artificial Intelligence (AI) makes a mathematical relation between the causes (faults) and the effects (sensor measurement and expert observations). This method is a procedure of mapping the information obtained in the measurement space and/or features in the feature space to system faults in the fault space. AI techniques have been increasingly applied to fault diagnosis and have shown improved performance over conventional approaches. In practice, however, it is not easy to apply AI techniques due to the lack of efficient procedures to obtain training data and specific knowledge, which are required to train the models. In the literature, two popular AI techniques for FDI are artificial neural networks and expert systems. Other AI techniques used include fuzzy logic systems, fuzzy-neural networks and Evolutionary Algorithms (EAs).

An artificial neural network is a computational model that mimics the human brain structure. It consists of simple processing elements connected in a complex layer structure which enables the model to approximate a complex non-linear function with multi-input and multi-output. A processing element comprises a node and a weight. The artificial neural network learns the unknown function by adjusting its weights with observations of input and output. This process is usually called training of an artificial neural network. There are various neural network models. Feedforward neural network structure is the most widely used neural network structure in system fault diagnosis [150, 151]. A cascade correlation neural network does not require

initial determination of the network structure and the number of nodes. The cascade correlation neural network can be used in cases where on-line training is preferable. Spoerre [152] applied cascade correlation neural network to bearing fault classification and showed that cascade correlation neural network can result in utilizing the minimum network structure for fault recognition with satisfied accuracy. Wang and Too [153] applied the unsupervised neural networks, self-organising map and learning vector quantization to rotating machine fault detection. Tallam *et al.* [154] proposed some self-commissioning and on-line training algorithms for feedforward neural network with particular application to electric machine fault diagnostics.

In contrast to neural networks, which learn knowledge by training on observed data with known inputs and outputs, expert systems utilize domain expert knowledge in a computer program with an automated inference engine to perform reasoning for problem solving. Three main reasoning methods for expert systems used in the area of fault diagnosis are rule-based reasoning, case-based reasoning and model-based reasoning [155].

Expert systems and neural networks have their own limitations. One main limitation of rule-based expert systems is combinatorial explosion, which refers to the computation problem caused when the number rule increases exponentially as the number of variables increases. Another main limitation is consistency maintenance, which refers to the process by which the system decides when some of the variables need to be recomputed in other values. Two main limitations of neural networks are the difficulty to have physical explanations of the trained model and the difficulty in the training process. Obviously, combination of both techniques would significantly improve the performance. Yang *et al.* [156] proposed an approach to integrate case-based reasoning expert systems with a neural network to enhance fault diagnosis. It was shown that the proposed approach outperforms the self-organising feature map-based system with respect to classification rate.

EAs, which mimic the natural evolution process of a population, have also been shown to have merits in applications to machine diagnosis. Genetic Algorithms (GAs) are the most widely used type of EA. Sampath *et al.* [157] proposed a GA-based optimization approach to gas turbine diagnosis. Several examples of other EA algorithms for fault classification and diagnosis are in [158, 159].

1.3.2 Related Works on Failure Prognosis

Today's concept of fault diagnosis comprises the automated detection and classification of faults, whereas failure prognosis the automated estimation of how soon and likely a failure will occur. Prognosis promises to significantly reduce expensive downtime, spares inventory, maintenance labor costs and hazardous conditions. However, prognosis is a relatively new research area and has yet to receive its prominence compared to the other areas of CBM [44]. Related reviews on prognostics have been reported in the literature [45-50]. The most widely used prognosis is to predict how much time is left before a failure occurs given the current system condition and past operation profile. The time left before observing a failure is called RUL. Generally speaking, failure prognosis techniques can also be classified into model-based and data-driven approaches.

Model based prognostic method

The model-based prognostic method utilizes an analytical model (dynamical model) to describe the system degradation for predicting the remaining time to reach a failure state. Compared with the data driven approach which requires costly data to train the prognosis system, the model-based technique may provide an economical mean to predict RUL. These models usually involve building technically comprehensive mathematical models to describe the physics of the system and failure modes, such as crack propagation in suspension, blockage of pipe in the hydraulic actuator and air leakage of tire in vehicle. These models attempt to combine system-specific mechanistic knowledge, defect growth formulas and CBM data to provide "knowledge-rich" prognosis output.

A common physics-based approach is crack growth modelling. Li *et al.* [51] proposed a recursive least squares based prognosis method for rolling element bearing using Paris-Erdogan law. The parameters in the model are tuned by an adaptation method based on features extracted from vibration signal. Least-square scheme enables adaptation of model parameters to changes in condition. Ray and Tangirala [52] used a non-linear stochastic model of fatigue crack dynamics for real-time computation of the time-dependent damage rate and accumulation in mechanical structures. An extended Kalman filter was used to predict the remaining service life.

However, the state equations for the dynamic system are not considered. Adams and Nataraju [53] modeled damage accumulation in a structural dynamic system as first/second order nonlinear differential equations. However, their damage state equations are functions only of the damage variables. The state variables (reflecting different usage) certainly influence the damage evolution. Chelidze *et al.* [54] modeled degradation as a “slow-time” process, which is coupled with a “fast-time” observable subsystem. The model was used to track the battery degradation of an electro-mechanical system. Qiu *et al.* [55] discussed the stiffness-based prognostic model for bearing systems, and this method is based on vibration response analysis and damage mechanics. Luo *et al.* [56] utilized an Interacting Multiple Model (IMM) to track the hidden damage. Remaining-life prediction is performed by mixing mode-based life predictions via time-averaged mode probabilities. The failure mode for this example involves a crack in the suspension spring caused by fatigue. This failure mode is linked to a suspension parameter (stiffness) that can be estimated (tracked) over time to assess the remaining life of the suspension subsystem. In this example, the post-action based on prognosis is to replace the suspension spring when the crack length reaches a predetermined critical length. Therefore, there are no interim maintenance actions.

A method to predict the RUL of a battery by model-based prediction approach is developed in [61]. The health of a battery can be indicated by a term named State Of Charge (SOC), which is defined as the ratio between the remaining capacity and the initial or rated capacity. To compute the RUL of a battery, an auto-regressive moving-average model is used to describe the behavior of the battery's SOC. Note that SOC is a slow varying term related to system damage. To compute the parameters in auto-regressive moving-average model, Randles circuit is used to model the battery so that the necessary parameters that are required for identifying the auto-regressive moving-average model parameters can be computed. Some failure modes of the SOC are passivation (battery does not accept charge), separation, bridging (battery is short-circuited), dryout, sulfation (the acid density is decreased), softening, that indicates a high discharge rate, thermal run-away (increase in battery temperature), and grid corrosion. A preliminary SOC prediction method has been proposed based on the auto-regressive moving-average model with the parameters trained for a certain failure mode. The battery is modelled based on its operating principles to make sure

the estimations are as close to the actual states as possible. Expert knowledge on the system is critical for model-based prognosis. Some other similar works on battery's SOC estimation can be found in [61-63].

In [124, 125], spur gear crack growth is modeled by Paris' law. A 2D finite element analysis model was integrated to calculate stress and strain fields based on gear tooth load, geometry and material properties. A stochastic version of the Yu-Harris bearing life equation is used to predict spall initiation and the Kotzalas-Harris progression model is adopted to estimate the time to failure [126]. In order to enhance the above method, a framework for physics-based prognostics which integrates material-level models, system-level data fusion algorithms and parameter tuning techniques is developed in [127]. Adams and Nataraju [128] proposed a method to model damage accumulation in a structural dynamic system as first-/second-order nonlinear differential equations.

A single fault growth model might not be often capable to capture a sequence of fault behaviors. Consider, for example, a rolling element bearing as a critical component of rotating machinery. The bearing may begin to corrode under certain operating conditions and, in parallel or sequentially, may be spalling and eventually, cracking. For accurate model-based prognosis, therefore, it is essential that fault progression models be developed to represent these evolving behaviors. An approach to multi-fault modeling of prognosis of a rolling element bearing of a helicopter's oil cooler is developed in [64]. A simple and cost-effective on-line parameter adaptation solution is introduced to improve the performance of modeling.

A different way of applying model-based methods to prognosis is to establish the explicit relationship between the condition variables and the lifetimes (current lifetime and failure lifetime) through mechanistic modelling. Two representative examples are [57] for machines considered as energy processors subject to vibration monitoring and [58] for bearings with vibration monitoring. Lesieutre *et al.* [59] developed a hierarchical modelling approach for system simulation to assess RUL. Engel *et al.* [60] discussed some practical issues regarding accuracy, precision and confidence of the RUL estimates.

The main advantage of the model-based method is its ability to incorporate a physical understanding of the system for health monitoring. Another advantage is that, in some situations, the changes in the feature vector are closely related to model

parameters. As a result, it can also establish a functional mapping between the drifting parameters and the selected prognostic features. Moreover, as the understanding of the system degradation improves, the model can be modified to increase its accuracy and to address subtle performance problems. Therefore, it can significantly outperform data-driven approaches.

Data driven prognostic method

Data-driven prognostic approaches attempt to derive models directly from routinely collected data instead of building models based on comprehensive system physics and human expertise. They are built based on historical records and produce prediction outputs directly in terms of data. These approaches usually make a mathematical relation between the causes (faults) and the effects (sensor measurement and expert observations) which is a procedure of mapping the information obtained in the measurement space and/or features in the feature space to system faults in the fault space. This mapping process is also called pattern recognition.

The conventional data-driven methods include simple projection models, such as exponential smoothing [118] and auto regressive model [119]. One major merit of these techniques is the simplicity of their calculations. However, most of these trend forecasting techniques assume that there is some underlying stability in the monitored system. They also rely on past patterns of degradation to project future degradation. Artificial neural network is the most commonly used data-driven method in the prognosis literature. Wang *et al.* [120] compared the results of applying recurrent neural networks and neural-fuzzy inference systems to predict the fault propagation trend in rotary machinery. The neural-fuzzy inference systems performed better than recurrent neural networks when the training data is sufficient. However, it could not provide reliable prediction if there was no sufficient training data or there were fast dynamic fluctuations. In order to overcome these problems, an adaptive training technique was proposed by Wang [121] to improve the neural-fuzzy model. Gebraeel *et al.* [122] developed a method to predict the actual bearing failure time instead of the future condition index considered in [120, 121].

In [136], a prognostic framework based upon the concept of dynamic wavelet neural networks is introduced. The dynamic wavelet neural network incorporates temporal information and storage capacity into their functionality so they can carry

out prognostic tasks for future. The prognostic architecture is based on a static “virtual sensor” that relates known measurements to fault data and a predictor which attempts to project the current state of the faulty component into the future, thus revealing time evolution of the failure mode and allowing estimation of the component’s RUL. Both constructs are based on a dynamic wavelet neural networks model acting as the mapping tool. A dynamic wavelet neural networks based degradation model that utilizes condition-based sensory signals to compute and continuously update residual life distributions of partially degraded components is developed in [123]. Initial predicted failure times are estimated through “supervised” trained dynamic wavelet neural networks using real-time sensory signals. These estimates are used to derive a prior failure time distribution for the component that is being monitored and update the prior distributions by using a Bayesian approach. The novelty of this methodology lies in the ability to update a component’s remaining life distribution using in situ condition-based sensory signals. The real-time sensory signals capture the latest degradation state of the component, and the resulting updated distributions are directly linked to the physical degradation state of the component.

Sheppard and Kaufman [137] developed a diagnostic approach based on Bayesian networks that incorporates information on failure probability, instrument uncertainty, and the predictions for false indication. They also used a dynamic Bayesian network, which is an extension of the Bayesian network, to perform prognosis by modeling changes over time. Bunks *et al.* first pointed out that Hidden Markov Models (HMM) based models could be applied in the area of prognostics in machining processes [138]. HMM is a parametric model; its parameters can be estimated by the vast experimental data using statistical techniques. HMMs have some distinct characteristics that are not possessed by some traditional methods. They could not only reflect the randomness of machine behaviors but also reveal their hidden states, changing processes. Furthermore, HMMs have a well-constructed theoretical basis and easy to realize in software. But HMMs have some inherent limitations. One is the assumption that successive system behavior observations are independent. The other is the Markov assumption itself that the probability in a given state at time t only depends on the state at time $t-1$ is clearly untenable in practical applications. Baruah and Chinnam presented a novel method to employ HMM for carrying out both diagnosis and prognosis [139]. The method applies HMM for modeling sensor signals and, in turn,

identifies the health states as well as facilitates estimation of RUL.

In [135], an on-line particle filtering based framework for fault diagnosis and failure prognosis is developed. This framework considers the implementation of two autonomous modules. A FDI module uses a hybrid state-space model of the plant and a particle filtering algorithm to estimate the state probability density function of the system and calculates the probability of a fault condition in realtime. Once the anomalous condition is detected, the available state probability density function estimates are used as initial conditions in prognostic routines. The failure prognostic module, on the other hand, predicts the evolution in time of the fault indicator and computes the RUL of the faulty subsystem, using a non-linear state-space model (with unknown time-varying parameters) and a particle filtering algorithm that updates the current state estimate. Data from a seeded fault test for a UH-60 planetary gear plate are used to validate the proposed approach.

PCA is usually used to extract the significant characteristics of structural failures out of the sensors. Its main function is to retain the most important characteristics of its inputs by using a small amount of data. The advantage of PCA is that it significantly reduces the dimension of input data and consequently enhances training speed and recognition speed. Furthermore, PCA has the self-learning capability.

PCA methods have been applied to many applications in image processing, signal processing, and pattern recognition. Recently, it has been introduced into the prognosis for the data preprocess purpose. Zhang *et al.* utilized PCA for the principal signal features extraction work that could help to generate a component's health/degradation index as the input of an on-line RUL prediction [140]. Kwan and Zhang employed PCA for the data extraction process in their fault diagnostics and prognostic model [141].

Data-driven methods have some advantages, such as capturing complicated phenomenon without a priori knowledge, and performing faster than traditional system identification techniques in multivariate prognosis for artificial neural network. However, the linkage between fault degradation and changes in physical parameters is generally lost with data-driven models. Moreover, it is a tough and high cost task to collect abundant data. Many algorithms and data training models require a large amount of historic data, including normal state data and failure (even fatal failure) data that need to destroy the components/systems artificially.

1.4 Problem Formulation

From the previous section, it is known there have been various efforts to hybrid system diagnosis. For instance, in [35], the HBG modeling technique is utilized to develop a qualitative-quantitative diagnosis framework for abrupt parametric faults in hybrid systems. In this work, the discrete events include known controlled mode changes and autonomous mode changes. However, if a fault is detected, the algorithm cannot track the autonomous mode changes. Moreover, when data is collected for fault parameter identification, the mode is assumed to be persistent. Since the mode changes may happen after fault occurrence, it is desirable that the fault estimation can be carried out when mode changes occur, in particular, the mode changes are unknown.

In some actual automatically controlled systems, faults can occur both on the mechanical part and on the electrical part of the system. In electromechanical systems like wind turbines, a single fault can occur on a specific actuator, on a specific sensor, or on a specific component of the system. The actuator and sensor faults are more frequent because of the presence of electrical devices and connections. Fault diagnostic algorithm considering only parametric faults is not enough for real application. Furthermore, if the system under monitoring is very complex and several faults may happen simultaneously. The fault diagnosis becomes difficult and a method that can deal with multiple faults is desired.

For some systems, e.g. a mobile robot, the fault could be the wearing of belt which leads to the loosen belt fault or a sudden disabling of driver that causes a burnt driver fault. In other word, the fault could be a sudden change of a component parameter or a gradual drift of a component parameter. In most cases, it is difficult to obtain the fault pattern information for a particular fault. Therefore, it is desirable to develop a method that is able to handle fault diagnosis without fault nature information.

Failure prognosis has received more attention recently, especially for continuous systems. However, the literature related to failure prognosis for hybrid systems is very limited. Prognosis for hybrid systems is challenging due to the fact that continuous dynamics and system mode are mutually dependent and intertwined. Mode information plays a vital role in designing prognosis algorithm for hybrid systems.

Traditional failure prognosis of continuous systems tracks the fault degradation and predicts the RUL for single component, while it cannot monitor several fault sources, such as components, sensors or actuators simultaneously. Therefore, prognosis of multiple failures is a challenging topic for system health monitoring.

1.5 Contributions

This thesis develops a quantitative model based diagnosis and prognosis framework for hybrid systems. In the framework, mode information is provided by mode tracker before fault occurrence. After a fault is detected, a unified formula is used to describe the mode changes. The developed algorithms can address multiple fault isolation without fault pattern and mode change information, simultaneous mode and fault parameter identification and prognosis of multiple failures. The main contributions of the thesis are as follows:

1. Augmented GARRs (AGARRs): are proposed to describe the fault in hybrid systems with parametric and nonparametric nature. The concept of AGARRs extends the capability of GARRs to concern faults of nonparametric nature using efficiency factors. The efficiency factor is defined to qualify the severity of the nonparametric fault and thus facilitate the fault identification and failure prognosis;
2. Parameterization of mode changes as a function of time, switching time stamps and initial mode coefficients to facilitate fault identification and prognosis. Before fault occurrence, the mode information can be provided by mode tracker in the QHBG framework. Once a fault happens, model based mode tracking technique is not reliable and can be misleading. Isolation of fault candidates leads to identification of fault parameters and evaluation of system health. However, if mode switching is unknown, the choice of a correct model is difficult for the identification purpose. With the help of the unified formula for mode changes, one can simultaneously identify fault parameters in fault candidates and switching time stamps to evaluate the severity of the faults without mode information;
3. Fault / mode change isolation: extend the hypothesis set generation to identify the simultaneous happening of mode change and fault as possible causes of the inconsistency. Most existing fault diagnosis methods cater exclusively to the faults occurring at the detectable mode. In real industrial hybrid systems, faults can

happen at a non-detectable due to ageing and wearing. These faults may be detected by FDI system if the system enters a new mode where the faults are detectable. The combinational effect of both mode change and the faults on residuals could lead to the observed inconsistency. Consequently, the fault hypothesis set during the fault isolation process will not only include suspected faults at current mode, but also the mode change and fault whose combinational effect on residuals may result in the inconsistency.

4. Prognosis of multiple failures: simultaneously track the degradation of multiple faulty components and predict the RULs for them. Most current works for prognosis only focus on single component degradation, such as a crack in bearing, battery state of charge, for continuous systems. Few works consider prognosis of multiple failures for hybrid systems. Consequently, prognosis for hybrid systems is a challenging topic due to the fact that system continuous dynamics and system mode (discrete state) are mutually dependent and intertwined. Mode information plays a vital role in design prognosis method for hybrid systems. A systematic prognosis method for hybrid systems is proposed in which the degradation model selection is discussed.
5. Fault diagnosis under different unknown fault patterns: the fault could be abrupt and incipient. The fault pattern of each fault is unknown in advance. Since in FDI systems, the nature of the fault is unknown beforehand, the fault could be sudden break of belt or slowly blockage of pipe. For sudden change of component parameter, it is often depicted as abrupt fault; whereas the slowly deteriorating of component is usually modeled as incipient fault. It is difficult to detect the incipient fault due to slowly developing nature of the fault and the compensation effect of feedback control. The situation is very harsh when the fault pattern information is unavailable, because it is hard to do fault identification which is based on the nature of the fault. In order to overcome this problem, the fault pattern selection is introduced as a part of fault identification task.
6. Validation through simulations and experiments on different hybrid systems. The proposed fault and mode switching identification method is studied using a complex electro-hydraulic suspension through simulation. The modes in the system have two types: one is autonomous mode changes which are governed by the values of internal state variables; another one is exogenous mode changes,

which are caused by external unknown events. The prognosis algorithm is studied using an electrical circuit which consists of many components. Several relays are employed to configure the system into different modes. Some degradation behaviors happen simultaneously in system components, sensors or actuators. An algorithm to realize prognosis of multiple failures is developed and different fault scenarios are studied. In the last part, a real world complex hybrid system – a CyCab mobile robot is tested using the proposed FDI method.

1.6 Organization of the Dissertation

The rest of this thesis is organized as follows:

Chapter 2 briefly reviews the modeling language, i.e. bond graph and hybrid bond graph, for model based health monitoring. Some issues, such as causality assignment for hybrid bond graph, monitoring ability analysis for hybrid systems and diagnostic hybrid bond graph, are addressed. Several computational intelligence methods related to hybrid system diagnosis and prognosis are also introduced.

Chapter 3 introduces the methodology of simultaneous fault parameter and mode change identification. The mode changes are parameterized as a function of time, switching time stamps and initial mode coefficients. Hence, after a fault is detected, the fault parameter value can be estimated without the requirement of mode change information. The developed method can deal with single parametric fault which happens at a detectable mode. Two examples, namely a complex electro-hydraulic suspension and an electrical circuit, are studied using the proposed algorithm.

Chapter 4 addresses the multiple faults detection, isolation and estimation without fault pattern and mode change information. The fault could be abrupt and incipient. The fault pattern of each fault is unknown in advance. These faults can develop at all modes, i.e. detectable modes and non-detectable modes. Multiple faults isolation without fault pattern information is discussed. A set of suspected mode/fault pattern is created after fault isolation module. For further refinement, fault estimation is utilized to evaluate the severity of the faults with correct pattern.

Chapter 5 develops a prognosis method of multiple failures which can simultaneously track the degradation of multiple faulty components and predict the RULs for those faulty components. Multiple incipient faults are considered in a complex hybrid system and these faults can develop during a mode when the faults are not detectable. The unknown degradation characteristic of each incipient fault is identified with closest matching one of some prescribed dynamic models. The resultant degradation model will serve as a base for prognosis. In the process of fault detection and isolation, the degradation models and faults are identified using a multiple adaptive hybrid particle swarm optimization algorithm.

Chapter 6 applies the QHBG based FDI framework in a hybrid system - a CyCab mobile robot test-bed. The system consists of many electrical, hydraulic and mechanical components. The faults under consideration include parametric fault, sensor fault and fault mode. For fault mode, the faulty state is known a priori, and can be modeled by known parameters only. An example of fault mode is the broken belt in a CyCab mobile robot where the model equations have to be restructured in order to describe the present faulty state of the system. The hardware structure and configuration are discussed and the experiment environment is introduced. The FDI decision making scheme is discussed. Three fault scenarios, i.e. an internal leakage fault, a pressure sensor fault and a burnt driver fault, are demonstrated. Online experiment results show the efficiency of the proposed method.

Chapter 7 presents the summary of the thesis and provides some possible future work related to the work done in this thesis.

Chapter 2

Preliminaries

This chapter introduces basic concepts of Bond Graph (BG) modeling methodology, hybrid system modeling technique and some useful results related to hybrid system diagnosis. Since the focus of this thesis is the integration of computational intelligence techniques with BG model-based diagnosis and prognosis, several computational intelligence methods are also discussed.

2.1 Bond Graph

Bond graph was invented by Henry Paynter at MIT in April 1959 [65] and subsequently developed into a systematic methodology by Rosenberg and Karnopp [161]. A bond graph is a graphical representation of a physical system. As a unified multi-energy domain modeling method, it provides an approach to model a complex system, which allows both a structural and a behavioral system analysis [66]. It provides a systematic and formal way for modeling dynamic systems with different energy domains, such as electrical, mechanical, hydraulic, etc., in a unified framework. BG is based on energy conservation law and allows for automatic generations of system equations. The BG model can also be used to obtain mathematical and graphical representation that becomes a foundation for monitoring ability analysis (ability to detect and to isolate faults) and supervision system design.

2.1.1 Bonds, Power and Causality

In BG theory, multiport refers to a point of view taken in the description of the subsystems. It represents any component (DC motor, pump, gearbox, etc.) designed to accomplish some specific objectives. Places at which subsystems can be

interconnected are places at which power can flow between subsystems. Such places are called ports, and subsystem with one or more ports is called multiport. A system with single port is called a 1-port, and a system with two ports is called a 2-port, and so on.

The device sketched in Figure 2.1 can be treated as two multiport elements (DC motor and pump) with ports that can be connected to each other to form a system. Figure 2.2 represents a word bond graph of the system. A word bond graph is a first step towards a bond graph, in which words define the multiport components. Figure 2.2 implies that a port of the motor, indicated by a single line emanating from the word representing motor, and a port of the pump have been connected, and hence a bond is formed. When two subsystems are physically connected together, a pair of complementary variables (called power variables) is constrained to be equal for the two subsystems. For the example in Figure 2.2, the motor and the pump have a common angular speed ω and torque τ at the coupling. The half arrow on the bond indicates the direction of power flow which is given by the product of the angular speed and the torque. When the product of the angular speed and the torque ($\omega\tau$) is positive, power is flowing from the motor to the pump. Similarly, power can be expressed as the product of a force and a velocity in mechanical domain, as the product of voltage and current in electrical domain. Since power interactions are always present when two multiports are connected, two power variables, effort (indicated by symbol e) and flow (indicated by symbol f) represent various power variables in different energy domains in a universal way. The product of these two power variables is the power $P = e \cdot f$.

Figure 2.1: Engineering multiport connection

Figure 2.2: Word bond graph for system of Figure 2.1

These two power variables in different domains are given in Table 2.1. Usually, flow represents either: current, flow rate or velocity; effort represents either: force, voltage or pressure. Bonds are numbered and the numbered effort and flow correspond to numbered bond. These analogies (which exist in different physical domains) enable the description of complex physical phenomena that integrate different physical domains and transformations of energy between different physical domains with a very small set of basic elements.

Table 2.1: Effort and flow in different domains [66]

Domain		Effort (e)	Flow (f)
Electrical		Voltage (e)	Current (i)
Mechanical	Translational	Force (F)	Velocity (v)
	Rotational	Torque (τ)	Angular Velocity (ω)
Hydraulic	Incompressible Flow	Pressure (P)	Volumetric Flow Rate (Q)
	Compressive	Enthalpy (h)	Mass Flow Rate ($\dot{m}$)
Thermal		Temperature (T)	Entropy Flow Rate ($\dot{s}$)
Magnetic		Magneto-motive Force (F)	Flux Rate ($\dot{\phi}$)

In bond graph theory, the way in which inputs and outputs are specified is by means of the causal stroke. The causal stroke is a short, perpendicular line made at one end of a bond. The casual stroke location indicates the direction of the effort, while the flow points in the opposite direction. The causality (cause and effect relationship) is a unique feature in bond graph theory. Thus, using A and B to stand for multiport subsystems in Figure 2.3, a concise bond graph notation with four pieces of information is given: the existence of link between two systems by the bond, the type of power (electrical, hydraulic, and so on) by the power variables e and f , the power direction by the half arrow and the causality by the stroke. In this example, e is the input to B and f is the output from B, while f is the input to A and e is the output from A. In short, bond graph consists of subsystems linked together by lines representing bonds. It is worth to note that half-arrow sign convention for power flow and the causal stroke are completely independent. For the example of Figure 2.3, the causal stroke may be put at the end near A, which implies that e is the input to A and f is the output from A.

Figure 2.3: A simple bond graph representation

2.1.2 Basic Bond Graph Elements

Using bond graphs and the classification of power variables, it turns out that only a few basic types of multiport elements are required to represent models in a variety of energy domains. In BG modeling, any energetic process can be described by generic BG components which include source elements Se and Sf , dissipative element R , storage elements C and I , two junctions 0 , 1 , and two transducers TF and GY .

2.1.2.1 Basic 1-Port Elements

A 1-port element is addressed through a single power port where a single pair of effort and flow variables exists. This type of element can be connected to another system through only one port. 1-port bond graph elements include passive 1-port elements and active 1-port elements. Passive 1-port elements either transform received power into dissipated power (element R), or store received power as kinetic energy (element I) or as potential energy (element C). For these passive 1-port elements, the power convention is established by means of a half arrow pointing toward the element as shown in Figure 2.4.

Figure 2.4: Representation of 1-port passive elements

The 1-port resistor (or resistance) is an R -element in which the effort and flow variables are related by a static function. Figure 2.5 shows a bond graph of an electrical resistor. In the figure, u is the supply voltage (effort variable) and i is the current (flow variable). The electrical resistor is an R -element with linear voltage-current constitutive relation according to Ohm's law $u= Ri$. Similarly, in mechanical domain the damper is considered as an R -element which is characterized by linear force-velocity relation.

Figure 2.5: 1-port resistor in electrical domain

In general, the constitutive equation of R -element can be represented as

$$\Phi_R(e, f) = 0 \quad (2.1)$$

In bond graph convention, the physical parameter of an element can be attached to the symbol of the element by a colon. For the case of Figure 2.5, R_e is the parameter of electric resistor which is attached to the 1-port resistor symbol R .

Next consider a 1-port device in which a static constitutive relation exists between an effort and a displacement (time integral of flow). This element is called a C -element (capacitor) which represents any system transforming the received power into potential energy without loss. Figure 2.6 shows a bond graph of spring in mechanical domain. In linear case, the constitutive law of the spring with stiffness k is

$$F = \frac{1}{C} \int \dot{x} dt = kx \quad (2.2)$$

where x is the relative displacement across the spring, $\dot{x}$ is the spring velocity, and F is the force. In this example, the compliance parameter $C = 1/k$.

Figure 2.6: 1-port capacitor in mechanical domain

Similarly, the electrical capacitor in electrical domain and the storage tank in hydraulic domain can also be considered as 1-port capacitors. The constitutive law for C -element which relates the effort to the time integral of flow is

$$\Phi_C(e, \int f dt) = \Phi_C(e, q) = 0 \quad (2.3)$$

where $q = \int f dt$ is displacement. The linear form of such constitutive law is

$$e = \frac{1}{C} \int f dt = \frac{q}{C}.$$

A second energy-storing 1-port element arises if the momentum p ($p = \int e dt$) is related by a static constitutive law to the flow f . Such an element is called inertia (I -element) in bond graph terminology. This element describes any system that transforms the received power into kinetic energy without loss. The I -element is used to model inductance phenomenon in electrical systems, mass or inertia effects in mechanical or hydraulic process. The 1-port inertia is characterized by the following constitutive law

$$\Phi_I(f, \int e dt) = \Phi_I(f, p) = 0 \quad (2.4)$$

If the relation is linear, the constitutive relation will have the form $f = \frac{1}{I} \int e dt = \frac{p}{I}$,

where I is the inertia parameter. An example of 1-port inertia in mechanical domain is depicted in Figure 2.7, the mass is ideally represented by the constitutive law

$$v = \frac{p}{M} = \frac{\int F dt}{M} \quad (2.5)$$

where v is the velocity, F is the force, M is mass of the block. The inertia parameter is M for this example.

Figure 2.7: 1-port inertia in mechanical domain

It is worth to note that passive 1-port elements can be modulated by external signal. For example, consider a valve controlled by ON/OFF controller. The volume flow rate across the valve will be zero when the valve is in OFF state. Under this case, the flow resistance is not a constant and is determined by the valve state, it is called modulated resistor. The same arguments also apply to I -element and C -element, and if these elements are not constant, they are called modulated inertia and modulated capacitor.

Finally, two active 1-port elements (effort source and flow source) are introduced. These sources are called active elements in bond graph terminology because they supply power to the process. Depending on the type of power variable the source provides, there are two 1-port source elements: effort source (Se) and flow source (Sf). Effort source imposes an effort on a system, which can be constant or a function of time, but independent of the flow variable. Effort sources include examples like voltage supplies, pressure pump, gravity, and so forth. As an example of a constant effort source, gravity force on a mass is essentially independent of the velocity of the mass near the surface of the earth. On the other hand, flow source provides a flow independent of the effort variable. Examples are electric current generator, imposed velocity on mechanics, and so on. Usually, source elements are thought of as supplying power to a system, so the bond is oriented toward the system connected to the source by convention. Figure 2.8 shows the bond graph representation of source elements.

Figure 2.8: 1-port active elements in bond graph

2.1.2.2 Basic 2-Port Elements

The 2-port elements introduced are ideal in the sense that power is conserved. There are two basic 2-port elements: transformer element, denoted by TF and gyrator element, represented by GY .

For a transformer element, the constitutive equations algebraically link inlet and outlet efforts and link inlet and outlet flows. The input effort is proportional to the output effort and the output flow is proportional to the input flow. The bond graph

representation of a transformer element is shown in Figure 2.9. In the figure, the subscripts 1 and 2 correspond to the two bonds. The constitutive law for a transformer element is given as

$$e_1 = me_2, f_2 = mf_1 \quad (2.6)$$

Figure 2.9: Bond graph representation of a transformer element where parameter m is called the modulus of the transformer. It is not hard to find that $e_1 f_1 = me_2 \times f_2 / m = e_2 f_2$. The sign convention of transformer is that one bond points toward the element and the other one is oriented away from the element.

Figure 2.10: A transformer element in mechanical domain

Figure 2.10 depicts an ideal rigid lever which represents a transformer element in mechanical domain. In the figure, F_1 and F_2 are the forces imposed on the ends of the lever and v_1 and v_2 are the velocities of the ends. a and b are length parameters. The modulus of the transformer of this example is b/a . The lever is an ideal transformer because kinetics indicates that $(b/a)v_1 = v_2$ and moment equilibrium requires $F_1 = (b/a)F_2$. This is exactly the definition for transformer element in (2.6). Other examples of transformers are pumps and cylinders in hydraulic domain, gear boxes in mechanical domain and electrical transformers in electrical domain.

Another energy conservative 2-ports bond graph element is gyrator, generally transforming energy from one domain into another. For a gyrator, the input effort is proportional to the output flow and the output effort is proportional to the input flow. The constitutive law of the gyrator is

$$e_1 = rf_2, e_2 = rf_1 \quad (2.7)$$

where r is the gyrator modulus. The general bond graph representation of a gyrator is depicted in Figure 2.11 where the subscripts 1 and 2 correspond to the two bonds. The power sign convention of gyrator is the same as the sign convention of transformer.

Figure 2.11: Bond graph representation of a gyrator element

Figure 2.12: An electrical gyrator element

An example of electrical gyrator is depicted in Figure 2.12, where u and i are supply voltage and current of the motor, ω and τ are output angular speed and torque, respectively. k_e is the current to torque ratio of the motor. The motor transforms electric power to mechanical rotary power. The constitutive law of this device is

$$u = k_e \omega, \tau = k_e i \quad (2.8)$$

The modulus of the gyrator of this example is k_e . Other examples of gyrators include voice coil transducer and gyroscope. If the transformer modulus m and the gyrator modulus r are not constant, but depend on time or any other parameter, then they are called Modulated Transformer (MTF) element and Modulated Gyrator (MGY) element, respectively.

2.1.2.3 Junctions

There are two types of connections represented in multiport forms, which in electrical terms, are called series and parallel connections. In bond graph terminology,

Figure 2.13: Bond graph representations of 1-junction and 0-junction

these connections are called 1-junction (common flow junction) and 0-junction (common effort junction). Figure 2.13 demonstrates the graph representations of 1-junction and 0-junction in bond graph theory. These two kinds of junctions are ideal in the sense that power is neither dissipated nor stored. The constitutive law of 1-junction and 0-junction can be expressed as

$$\sum_{i=1}^n e_i f_i = 0 \quad (2.9)$$

where n is the totally number of bonds connected to the junction and Σ is the algebraic sum. The sign is taken (+) when the power is oriented towards the junction and (-) when it is outwards from the junction. Consider the model in Figure 2.13 (a), bond 1 and bond 4 have power directions towards the junction and the power directions of bond 2 and bond 3 are out of the junction, thus the constitutive law is represented as

$$e_1 f_1 - e_2 f_2 - e_3 f_3 + e_4 f_4 = 0 \quad (2.10)$$

The 1-junction is defined such that all flows are the same, thus

$$f_1 = f_2 = f_3 = f_4 \quad (2.11)$$

Combining (2.10) and (2.11) yields

$$e_1 - e_2 - e_3 + e_4 = 0 \quad (2.12)$$

In summary, the flows on all bonds of 1-junction are always identical, and the algebraic sum of the efforts is equal to zero.

Figure 2.14: An example of bond graph representation involving 1-junction

An example of bond graph model involving 1-junction is depicted in Figure 2.14. In the figure, k is the spring stiffness, b is the damping coefficient of the damper and M is the mass of the cart. F is the force imposed on the cart and $\dot{x}$ is the velocity. In the corresponding bond graph model, the input force is modeled by the effort source Se , the spring is modeled as C -element with $C=1/k$, the damper is modeled as R -element with $R=b$, and the cart is modeled by I -element with $I=M$. F_C , F_R and F_I are forces imposed on the spring, damper and mass, respectively. A common velocity exists among these elements and a dynamic equilibrium of forces indicates that the algebraic sum of the efforts is equal to zero.

For 0-junction, the associated bonds share the same effort, and the algebraic sum of the flows always vanishes. The constitutive equations in Figure 2.13 (b) can be written as

$$\begin{cases} f_1 - f_2 - f_3 + f_4 = 0 \\ e_1 = e_2 = e_3 = e_4 \end{cases} \quad (2.13)$$

Figure 2.15: An example of bond graph representation involving 0-junction

Figure 2.15 depicts an example of bond graph model involving 0-junction. It is obvious that the voltage source, R -element and C -element share the same effort (voltage) and the sum of currents is equal to zero according to Kirchhoff's current law.

2.1.2.4 Sensors

For the control of engineering systems, sensors are necessary components to measure signals of the system responses, e.g., velocities, torques, or temperatures. Usually, sensors perform a conversion of a non-electrical signal of low power into an electrical one, which is converted into a digital signal. A characteristic feature of sensors is that they sense a signal without affecting the system. The amount of power they take out of the system is very small and can be neglected. For example, an ideal ammeter indicates current but introduces no voltage drop, an ideal voltmeter reads a voltage while passing no current, and the like. In bond graph model, the signal measured by a sensor is represented by a full arrow bond.

Figure 2.16: Bond graph representations of sensors

Figure 2.16 depicts the bond graph representations of effort sensor (De) and flow sensor (Df). The causal stroke location decides the type of the sensor, where the causal stroke is located near the sensor element for effort sensor and the causal stroke is put away from the sensor element for flow sensor. The full arrow bond is also called information bond where one of its conjugate power variables is set to zero. For example, the effort sensor provides only information about the effort measurement, such as pressure, force, voltage, and so forth. There is no reaction from the sensor to the system. Thus the flow of the effort sensor is zero.

2.1.3 Causality Consideration for Basic Bond Graph Elements

Bond graph has the notion of causality, indicating which side of a bond determines the instantaneous effort and which determines the instantaneous flow. This part concerns some more specific constraints of causalities in basic bond graph elements.

2.1.3.1 Causality for Basic 1-Port Elements

For bond graph source elements, the causalities are depicted in Figure 2.17. In the figure, the location of causal stroke indicates the direction that the effort variable is oriented.

Figure 2.17: Bond graph causality assignment of sources

In bond graph theory, the 1-port resistor is normally indifferent to the causality imposed on it. Figure 2.18 shows the two possibilities of causalities assigned to resistor. Figure 2.18(a) represents the resistive causality of resistor where the input is flow and output is effort, while Figure 2.18(b) denotes the conductive causality of resistor where input is effort and output is flow.

Figure 2.18: Bond graph causality assignment of resistor

These two possibilities can be represented in equation form as follows

$$e = \Phi_R(f) \quad (2.14)$$

$$f = \Phi_R^{-1}(e) \quad (2.15)$$

where (2.14) represents constitutive law of the resistive causality of resistor, (2.15) represents constitutive law of the conductive causality of resistor.

As for energy-storing 1-port element C , there are two types of causality, namely integral causality and derivative causality as shown in Figure 2.19. For integral causality of C -element, f is the input and e is given by a static function of time integral of f

$$e = \Phi_C^{-1} \left(\int f dt \right) \quad (2.16)$$

and for derivative causality, e is the input, f is the time derivative of a static function of e

$$f = \frac{d}{dt} \Phi_C(e) \quad (2.17)$$

Figure 2.19: Bond graph causality assignment of capacitor

Another energy-storing 1-port element is I -element, similarly, it has integral causality and derivative causality as depicted in Figure 2.20.

Figure 2.20: Bond graph causality assignment of inertia

In contrast to C -element, the input is e for integral causality of I -element and derivative causality exists when f is the input variable. These two causality representations can be written as

$$f = \Phi_I^{-1} \left(\int e dt \right) \quad (2.18)$$

$$e = \frac{d}{dt} \Phi_I(f) \quad (2.19)$$

2.1.3.2 Causality for Basic 2-Port Elements

For transformer element, denoted by TF , in bond graph theory, there are two possible causality assignments as shown in Figure 2.21. The constitutive law of the transformer in Figure 2.21 (a) can be represented as

$$e_1 = m e_2, \quad f_2 = m f_1 \quad (2.20)$$

The constitutive law of the transformer in Figure 2.21 (b) can be represented as

$$f_1 = f_2 / m, \quad e_2 = e_1 / m \quad (2.21)$$

Figure 2.21: Bond graph causality assignment of transformer

For a gyrator, the causality assignments are depicted in Figure 2.22.

Figure 2.22: Bond graph causality assignment of gyrator

For Figure 2.22 (a), the constitutive law is

$$e_1 = rf_2, \quad e_2 = rf_1 \quad (2.22)$$

The constitutive law of the gyrator in Figure 2.22 (b) can be represented as

$$f_1 = e_2 / r, \quad f_2 = e_1 / r \quad (2.23)$$

2.1.3.3 Causality for Junctions

For 1-junction, all flows on all the bonds are equal and the efforts must sum to zero. If the flow on any single bond is an input to the junction, the flows on all other bonds are determined. The bond that determines the flow is called the strong bond of 1-junction and this bond has causal stroke away from the junction. All other bonds are called weak bonds. The output of the 1-junction is the effort variable of the strong bond and the effort of the strong bond can be expressed as signed sum of efforts of other bonds. The causality for 1-junction is shown in Figure 2.23. In the figure, bond 1 is the strong bond and all other bonds are weak bonds. For 1-junction, there is only one bond which has causal stroke at the end away from the junction and other bonds have causal strokes at the end near the junction.

Figure 2.23: Bond graph causality assignment of 1-junction

Figure 2.24: Bond graph causality assignment of 0-junction

Consider 0-junction, there is only one bond which has causal stroke at the end near the junction and other bonds must have causal strokes at the end away from the junction. The bond that decides the effort of the junction is called the strong bond of 0-junction and it is causalled near the junction. All other bonds are weak bonds of 0-junction. The output of the 0-junction is the flow variable of the strong bond and the flow variable of the strong bond can be expressed as algebraic sum of flows of other bonds. Figure 2.24 depicts the causality for 0-junction where bond 1 is the strong bond and other bonds are weak bonds.

2.1.4 Sequential Causality Assignment Procedure

Besides the generalized modeling capability, BG modeling also utilizes a concept of the causality to derive equations from the graph. Every bond of the graph has a stroke marked at one end of the bond. For a bond in BG, the causal stroke location indicates the direction of the effort, while the flow points in the opposite direction. This indication allows users to derive relations between system's variables systematically. Causal path is a graphical representation that explicitly shows the relation between system variables based on the concept of causality. In bond graph theory, when a signal (effort or flow) reaches a junction, it is either distributed (flows any one of the weak bonds if the signal is the variable of the strong bond of the junction) or goes to the strong bond (if it is a variable of the weak bond). When the signal reaches a passive element (R , I or C) it returns back through it but undergoes a

qualitative change (flow variable becomes effort variable and vice verse). When a signal reaches a source or sensor, it is terminated there.

Figure 2.25: An example of a BG with causality

Figure 2.25 illustrates an application of using causality to derive an analytical expression of a variable systematically. In the figure, there are three components Se , R and 1-junction. The causality assigned to the graph is represented by the strokes at the end of bond 1 and bond 2. Consider the case where it is desirable to find the analytical equation of variable f_2 . For the R component in the figure, e_2 is the input and f_2 is the output. This implies that the constitutive relation (Φ_R^{-1}) of R is $f_2 = \Phi_R^{-1}(e_2)$. Similarly, the causal stroke of bond 1 reveals that $e_2 = e_1 = Se$. Hence, the analytical connection leads to $f_2 = \Phi_R^{-1}(Se)$. Note that the analytical relation between f_2 and Se can be shown explicitly using causal paths as shown in the figure. This illustration shows that BG provides a method to derive relation of a variable analytically and systematically. This advantage would be more significant for a large complex system.

An energy storage element (C or I) is independent if it used integral causality. If an energy storage element is assigned with derivative causality, then the element is not independent, and its energy variable (displacement or momentum) is algebraically related to the other energy variables in the system. Such energy storage element in derivative causality still store energy, but it does not contribute a state variable (displacement or momentum) to the system. Thus, in causality assignment, it is preferred to assign integral causality to each energy storage element. Another reason choosing integral causality over derivative causality for energy storage element is the numerical problem in computer simulation because the output of an energy store element with derivative causality exhibits a pulse of infinite height if the input immediately jumps to another value at certain time. This problem does not appear

with integral causality. A procedure, named Sequential Causality Assignment Procedure (SCAP) was developed to assign these causalities to a BG [66]. The procedure can be described as follows:

1. Choose any source (Se or Sf), and assign its required causality. Immediately extend the causal implications through the graph as far as possible, using the constraint elements ($0, 1, TF, GY$).
2. Repeat Step 1 until all sources have been assigned.
3. Choose any energy storage element (C or I), and assign its preferred integral causality. Immediately extend the causal implications through the graph as far as possible, using the constraint elements ($0, 1, TF, GY$).
4. Repeat Step 3 until all energy storage elements have been assigned.
5. Choose any unassigned R -element and assign a causality to it (basically arbitrary). Immediately extend the causal implications through the graph as far as possible, using the constraint elements ($0, 1, TF, GY$).
6. Repeat Step 5 until all remaining bonds have been assigned.

An example in electric domain is shown in Figure 2.26. Causality assignment is carried out step by step in the example using SCAP. In Figure 2.27(a), bond 1 is directed according to the source element Se . Since the strong bond for the 1-junction has not decided, and other bonds cannot yet be assigned a causality. There are no more sources, and then the process proceeds to Step 3. In Figure 2.27(b), bond 2 and bond 5 are causally directed to produce integral causalities for I -element and C -element. Thereafter, bond 3 is directed using the constraint of 1-junction.

Figure 2.26: A RLC electric circuit

Figure 2.27: Causality assignment process of the electric circuit

Finally, bond 4 is causally according to the constraint of 0-junction. The complete bond graph model is depicted in Figure 2.27(b). From Figure 2.27(b) with proper causality assignment, one can understand that the components C and R have the same effort, i.e. are connected in parallel using 0-junction. L and V_{in} share with the same flow, since they are connected in series. It can also be seen from the diagram, how the power of the source V_{in} is distributed between the different components.

2.2 Hybrid Bond Graph

Figure 2.28: Diagram of hybrid systems

Hybrid systems consist of continuous dynamics and discrete behavior represented by modes. Figure 2.28 shows the diagram of hybrid systems. Discrete transitions force the system to undergo from one mode to another. In each mode, the system is governed by continuous dynamics, and different modes correspond to different continuous models. There are various efforts to model the discontinuous behavior in hybrid systems. In [70], a modulated transformer is developed to model the discrete behavior in hybrid systems. When the switch is closed, the modulation parameter m for the modulated transformer is 1, while m takes value of 0 when the switch is on. However, when the switch opens the transformer connected junction, incorrect flow values will present. Sometimes this junction fails to disconnect the subsystems due to the existence of freewheeling flows. Moreover, causality reassignment is required when the switching occurs. In order to overcome this problem, a combination of

modulated transformer and resistance R is proposed in [71], where a fixed causality BG is achieved even if the physical configuration changes. However, the presence of R gives rise to dissipation, which indicates a non-ideal switching. In addition, resistance results in numerically stiff systems that will increase the simulation time. An ideal switch is proposed in [72] as a new BG element. The element enforces zero effort on the junction when the switch is ON and zero flow when the switch is OFF. Since effort or flow is always zero, power (effort $\times$ flow) associated with the switch element is always zero. The disadvantage of this approach is that one needs reassign the causality if switching happens. To handle this problem, a concept of switched power junction is developed to represent switching phenomena in systems [73, 74]. The switched power junction models ideal switching, so the problem of stiff systems and associated numerical stability problems while simulating the system are eliminated. The SJP are a generalization of the standard 0 and 1 junctions. Thus, the 0 and 1 junctions are special cases of the more general switched power 0 and switched power 1 junctions. BG is geared towards modeling of continuous systems. To model a hybrid system using BG language, additional features are necessary to capture the discrete mode changes of the hybrid system.

Figure 2.29: A simple hybrid system: electrical circuit

Figure 2.30: HBG model of the circuit

when the switch is ON

Figure 2.31: HBG model of the circuit

when the switch is OFF

HBG extends BG modeling by incorporating the controlled junctions to enable the hybrid system to be modeled using the BG components [67-69]. For 1-type controlled junction, it enforces zero flow to all connected bonds when it is turned off; similarly, a

0-type controlled junction forces the effort value to zero at all connected bonds during the OFF state. Consequently, the hybrid systems can be represented by HBG in a compact manner. An example of hybrid system, i.e. electrical circuit, is depicted in Figure 2.29, where a switch (S_w) is used to control the ON/OFF of the hybrid system. The HBG model of circuit is presented in Figure 2.30, where junction 1_c denotes the controlled junction. When the controlled junction is ON, it functions like standard 1-junction as shown in Figure 2.30; when the junction is OFF, it enforces zero flows to all adjacent connected bonds, then bond 1, 2 and 3 are deactivated and this state is represented in Figure 2.31 by dash lines.

2.2.1 Causality Properties and Causality Assignment for HBG

HBG is a BG-based modeling approach which provides an avenue to model complex hybrid systems. However, due to mode-varying causality properties of HBG, HBG has not been efficiently-exploited for fault diagnosis. In this part, a comprehensive study on the HBG from FDI viewpoints is presented. Some properties pertaining to the HBG are introduced. Based on these findings, a causality assignment procedure technique is developed to achieve a HBG with a desirable causality assignment that leads a unified description of system's behavior. These results lay a foundation for quantitative FDI design for complex hybrid systems.

2.2.1.1 Causality Properties of HBG

HBG extends the ability of BG to model hybrid systems using controlled junctions. However, changes in configuration through the various operating modes of the system can result in reassignment of causality. In [76, 77], the properties of HBG are studied so that it can be efficiently utilized for simulations of hybrid systems. A hybrid SCAP algorithm is developed to reassign causality incrementally after mode changes, starting from the junction directly affected by the switching, and then propagating the changes only to those junctions whose causal assignments were affected. In [77], this procedure is improved by using the concept of junctions in fixed causality and bonds with persistent causality. To exploit the rich information of a HBG for FDI, it is desirable to have a consistent causality description for the hybrid system at all modes. This consistent causality description facilitates FDI designs by eliminating the need

for causality reassignment which leads to derivation of unified constraint relations.

Figure 2.32: A hybrid system of electrical circuit

Figure 2.33: HBG of with a causality assignment when the switch is OFF

Figure 2.34: HBG with an alternate causality assignment when the switch is OFF

Let us consider a simple hybrid system in electrical domain with two modes, and the mode is determined by the state of an electrical switch as depicted in Figure 2.32. Two HBG models with different causality assignments are depicted in Figure 2.33 and Figure 2.34; for reference purposes, all the bonds and BG components of the HBG are enumerated. The full arrow bond for switch sw_1 means that a signal flow occurs at essentially zero power flow. Note that the purpose of the BG models considered in this work is for diagnosis, so the storage elements are assigned preferred derivative causality [75]. This is because ARR equations that are derived based on derivative causality are free from unknown initial state. In the figures, the active bonds are represented by solid lines while inactive bonds are represented by dash lines.

Here active bonds mean those bonds which are not deactivated by an OFF state of controlled junction.

Figure 2.33 describes three inactive bonds during the OFF state of the controlled junction 1_1 . Notice that the inactive bonds cause invalid causality to junction 0_2 which leads to the invalid causality of the HBG during the OFF state. This invalid causality obstructs the generation of the constraint equations at the OFF mode and the problem is significant for complex systems with many controlled junctions. Next consider the same HBG with an alternate causality assignment as shown in Figure 2.34. In the figure, the inactive bonds during the OFF state of the controlled junction 1_1 don't pose any invalid causality to the active BG components. The junction 0_2 still has a proper causality when the bond 3 is deactivated. This observation suggests that it is possible to describe the behavior of a hybrid system based on one unified causality assignment where all active bonds' causalities are consistent, and the causality is valid at all modes.

Without loss of generality, the general case where a controlled junction is connected to a junction $\{0, 1\}$, a 1-port component, or a 2-port component is considered. First, the case of BG junctions is considered.

Figure 2.35: 1-controlled junction connected with 0-junction

Figure 2.36: 0-controlled junction connected with 1-junction

Figure 2.35 depicts a 1-controlled junction (1_c) and a BG junction with the given causality assignment shown in the figure. The causal stroke of bond 3 indicates that the effort (e_3) is the output variable of 1_c , which is also an input variable of the 0-junction; where f_3 is the output variable of 0-junction and an input variable of 1_c . By inspection, if the controlled junction is OFF, the causality of the 0-junction becomes invalid. This behavior is due to the OFF state of 1_c , which causes the output variable of the 0-junction to be undefined. Similarly, this behavior also applies to a 0-controlled junction, as shown in Figure 2.36. This property is stated as follows.

Property 1: Suppose a controlled junction is adjacently connected to a normal junction $\{0, 1\}$. If the output variable of the controlled junction is an input variable of the normal junction, then the OFF state of the controlled junction poses an invalid causal form to the normal junction.

In HBG language, all 1-port components that are adjacently connected to a controlled junction are disconnected from the remaining graph when the junction is OFF (i.e. their single bond is deactivated). Therefore, consideration of causal validity of any 1-port component is not required.

Property 1 provides a causal condition on the controlled-junctions which is needed to be avoided when assigning the causality of the HBG. Besides property 1, a condition that the OFF state of the controlled junction poses no invalid causal form to the HBG can be deduced. One such condition can be stated as follows.

Property 2: If the output variable of a controlled junction is an input variable of a 1-port component, then the OFF state of the controlled junction poses no invalid causal form to the HBG.

To show this property, let us consider the following cases: 1) a controlled junction that is adjacently connected to a normal junction and 2) a controlled junction that is adjacently connected to a 2-port element. First, let us consider case 1). In BG language, any two adjacently connected junctions of the same type can be merged into one junction; therefore, it is assumed that any junction which is adjacently connected to a 0-controlled junction is a 1-junction. From the rules of causality, if the output variable of a 0-controlled junction is an input variable of a 1-port component, then the output variable of the 1-junction is not a variable of the bond that connects between the two junctions. Therefore, the output variable of the 1-junction is always well-defined, i.e., the 1-junction always has a valid causal form even when the 0-controlled junction is OFF. Now, consider case 2) where the controlled junction is adjacently connected to a 2-port component. The causality rules of 2-port component imply that when the 0-controlled junction is OFF, it imposes zero effort at the effort-input of the 2-port component (i.e., the input from the controlled-junction side) and the constitutive relation of the 2-ports component remains valid. This validity of the 2-port constitutive relations implies that the component has a valid causal form even when the controlled junction is OFF. Therefore, the OFF state of the controlled junction poses no invalid causal form to the HBG. The same argument applies to the case of 1-controlled junction.

Property 2 states a sufficient condition for a controlled-junction's causality that will not cause an invalid causal form to the HBG when the junction is OFF. One critical implication of this sufficient condition is that if the causalities of all controlled junctions are assigned in such a way that all their output variables are inputs of some 1-port components, then all active bonds maintain their causal forms, and the causality of the graph is valid at all operating modes.

One intuitive property can be useful in a special case is presented as follows.

Property 3: Consider two adjacently connected controlled junctions that have the same states at all operating modes. If one of the controlled junction's output variable is an input variable of the other controlled junction, then the OFF states of the controlled junction pose no invalid causal form to the HBG.

Property 3 is straightforward because the output variables of the two controlled-junctions are not input variable of any normal BG junction that is adjacently connected to them, and hence the OFF states of the controlled-junctions pose no invalid causal form to the HBG. Moreover, since the two controlled-junctions always have the same states, the OFF state of one implies the OFF state of the other; therefore, the two controlled-junctions do not pose any invalid causal form on themselves.

2.2.1.2 Causality Assignment for HBG

From the properties mentioned earlier, it is concluded that a controlled junction is said to be in preferred causality for FDI if the junction's output variable is an input variable of the following components: a 1-port component (R , C or I); a source element if the source is null when the junction is OFF; another controlled junction of different type which shares the identical state; or a 2-port component (TF or GY) with two controlled junctions at its both side having identical state. For a given acausal HBG (HBG without causality assignment), it is desirable to assign a causality to the HBG such that all controlled junctions are in preferred causalities. In this work, the focus is on FDI applications; hence, it is desirable that all the storage components (C and I) of the HBG to be in derivative causalities. This is because ARR equations that are derived based on derivative causality are free from unknown initial state. A HBG with such causality assignments is named as Diagnostic Hybrid Bond Graph (DHBG). In short, a DHBG is a HBG with a causality assignment that all its controlled junctions and storage components are assigned with preferred causalities. Note that integral causality is recommended for engineering simulation for ease of formulation and avoiding the numerical problems from differentiation. From the FDI perspectives, it is beneficial to design FDI algorithms based on a DHBG since the consistent causality of the graph provides a convenient way to describe the behavior of the hybrid system at all modes. Here, a systematic procedure called the Sequential

Causality Assignment Procedure for Hybrid Systems (SCAPH) is developed to achieve a DHBG from an acausal HBG [40].

The procedure is listed as follows:

- 1) Choose any controlled junction that has no source component adjacently connected to it and assign the junction with its preferred causality.
- 2) Repeat Step 1) until all the controlled junctions that have no source component connected to them have been assigned.
- 3) Choose any remaining controlled junction that has a source component connected to the junction. For the pair $\{S_e, 1_C\}$ or $\{S_f, 0_C\}$, the output variable of the controlled junction must not be an input variable of $\{0, 1\}$. For the pair $\{S_e, 0_C\}$ or $\{S_f, 1_C\}$, the output variable of the controlled junction must be the input variable of the connected source component.
- 4) Repeat Step 3) until all the remaining controlled junctions have been assigned. After this step, all controlled junctions are treated as normal junctions $\{0, 1\}$ in the following procedures.
- 5) Choose any remaining source component and assign its causality. Immediately extend the causal implications through the HBG as far as possible using the constraint components $\{0, 1, GY, TF\}$.
- 6) Repeat Step 5) until all sources have been assigned.
- 7) Choose any storage component and assign it with a derivative causality. Immediately extend the causal implications through the HBG as far as possible using the constraint components $\{0, 1, GY, TF\}$.
- 8) Repeat Step 7) until all storage components have been assigned with a causality.
- 9) Choose any unassigned R component and assign a causality to it. Immediately extend the causal implications through the HBG as far as possible using the constraint components $\{0, 1, GY, TF\}$.
- 10) Repeat Step 9) until all remaining bonds have been assigned.

2.2.2 Global Analytical Redundancy Relations

In order to check structural observability directly on a causal bond graph, Sueur and Dauphin-Tanguy have given the following necessary and sufficient criterion [160].

A system is said to be observable if and only if the following two conditions are satisfied.

1. Given that the preferred causality applied to the energy storage elements in the bond graph is integral causality, then every energy storage element in integral causality must have at least one causal path to a sensor.
2. In a bond graph with preferred integral causality, all energy storage elements in integral causality must take derivative causality if derivative causality is the preferred causality. If this condition is not met directly, then it is achieved by replacing some sensors in appropriate places by their dual, where the causalities of these sensors should be inverted.

Since in the DHBG, all storage components are assigned with derivative causalities according to SCAPH. The hybrid system with DHBG for FDI purpose is observable. It is worth to note that the main goal of the SCAPH is to assign the HBG a desirable causality that facilitates analysis and FDI designs. It can be observed that the preferred causalities of the controlled junctions impose additional restrictions to the causality assignment of the HBG; as a result, it is possible that some controlled-junctions may not be assigned with their preferred causalities due to the restrictions. Under this case, the DHBG can be achieved by SCAPH with allowing inversion of sensor causalities when necessary.

A model-based FDI approach evaluates the system's constraint using known information, e.g., known inputs, sensors data, and parameter values of the monitored system. These constraint relations are called the Analytical Redundancy Relations (ARRs). ARRs are static or dynamic constraints representing relations between different process variables. In essence, ARRs are combination of system's observations which are expressed only in terms of known variables. ARRs verify the consistency of the monitored system during operation. Any inconsistency in one of these relations is an indication of a fault which occurred in the monitored system. The numerical values evaluated from the ARRs are called residuals. Besides fault diagnosis, the ARRs can also be used for offline structural analysis such as fault detectability and fault isolability of each component under consideration [80-82]. The parity space method is a well-known ARR based FDI technique, but the way to derive ARR is different with the one used in BG based FDI [165, 166]. The parity space is well suited for sensors and actuators FDI [164]. While the identified model is given

under state equation form, the generated ARR parameters do not have a physical perception and cannot be associated with physical component fault. In other words, by comparing a BG based FDI with a parity space FDI method using state-space model, BG provides component-level information that allows the FDI algorithm to isolate a fault at component level easily.

ARR generation is carried out by eliminating the unknown variables in the system constraint equation. However, eliminating these unknown variables is not a trivial task, especially for complex systems. For linear systems, the parity space technique in [83] can be treated as elimination of the unknown variables using projection method [84]. Similarly, this concept has been extended to nonlinear systems for generation of ARRs [85]. Nevertheless, for large complex systems, the elimination of these unknown variables becomes a daunting task. Fortunately, BG allows users to deal with large amount of equations by exploiting the causal paths of the graph. Causal paths are graphical representations that are determined from the causality assigned to the BG. The paths explicitly show the relations between variables. Since an ARR is derived by solving unknown variables, the ARR can be derived systematically based on the causal paths. For continuous systems, BG has proved itself a useful tool for generating ARRs for complex systems [78-82].

Based on the uniform causal paths in DHBG model of the system, a set of unified relations, termed Global Analytical Redundancy Relations (GARRs), is derived to describe the hybrid system at all modes. Similar to continuous systems fault diagnosis, the GARRs provide a convenient tool to deduce the fault detectability and isolability of a hybrid system. In general, GARR equations take the following form

$$g_l(\theta, \alpha, De, Df, u) = 0 \text{ for } l = 1, 2, \dots, m \quad (2.24)$$

where

- m number of GARRs derived from the DHBG;
- θ nominal parameters of the system, $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$;
- α mode switching states of the q controlled junctions, $\alpha = [a_1, a_2, \dots, a_q]^T$;
- u system input;
- De, Df effort and flow sensors.

The procedure of deriving the GARRs from DHBG is listed as follows [39]:

- 1) Define a boolean variable a_i that represents the state of each controlled junction $i = \{1, \dots, q\}$, and let N be the total number of sensors modeled in the DHBG.
- 2) Let n_i be the total number of bonds of the controlled junction i . For each controlled junction i , there are n_i bonds adjacently connected to it. Every flow variable f_i^j (for $j = 1, \dots, n_i$) of the n_i bonds is substituted with $a_i f_i^j$ if the considered junction is 1-Type. Similarly, every effort variable e_i^j (for $j = 1, \dots, n_i$) of the n_i bonds is substituted with $a_i e_i^j$ if the considered junction is 0-Type.
- 3) Repeat Step 2 until all the controlled junctions are considered.
- 4) Choose a BG component. The component can be a junction, controlled-junction or any other BG component (such as 1-port or 2-ports component).
- 5) Based on the constitutive relations of the considered BG component, all unknown variables of these relations are to be eliminated using the causal paths of the DHBG. If all unknown variables of any of these relations are solved, i.e., expressed only in terms of known variables (input, sensors variables), then the GARR is the relation with its solved unknown variables. Repeat solving until all the considered constitutive relations have been attempted.
- 6) Choose another BG component.
- 7) Repeat Step 5 until all the components of the DHBG have been considered, or until the number of generated GARRs equals N , where N is the total number of sensors.

For a continuous system, a residual r_l , defined by a ARR, is sensitive to a fault in those components whose parameters appear in the ARR. When the system is fault-free, all residuals are consistent with the system's behavior and residuals $|r_l|$ are zeros or below a small threshold ε_l . To apply these residuals for FDI, a binary coherence vector $C = [c_1 \dots c_m]$ is defined. Each component c_l of C is defined as follows:

$$c_l = \begin{cases} 1 & \text{if } |r_l| > \varepsilon_l \\ 0 & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \dots, m \quad (2.25)$$

When the system is fault-free, the binary coherence vector C will be zero. On the other hand, if the system is faulty, then at least one entry of the coherence vector will take value 1. Fault monitoring ability of a system can be studied using a Fault Signature Matrix (FSM) generated from the m ARR equations [90-93, 95]. In the

FSM, column headers consist of residuals $r_1, \dots, r_m$, fault detectability (D_b), and fault isolability (I_b). Each entry of the table holds a Boolean value. For each row, the Boolean entries under the columns $r_1, \dots, r_m$ form the fault signature of the parameter θ_i corresponding to a fault in the parameter θ_i . Under a residual column, a value 1 in an entry indicates that the residual is sensitive to a fault in the parameter of the matching row. On the other hand, a 0 in the entry represents the residual is insensitive to the fault in the corresponding parameter. If at least a 1 appears in the fault signature of a parameter θ_i , then the parameter is said to be fault detectable. This ability is represented by a $D_b = 1$ in the matrix. When the fault signature of a parameter θ_i is unique, then a fault occurred in parameter θ_i is said to be fault isolable. This is denoted by $I_b = 1$. Unlike continuous systems, hybrid systems have multiple modes but only one mode is in operation at any time. This suggests that it is necessary to evaluate the system's monitoring ability in different operating modes for FDI analysis and designs. By design, the global characteristic of the GARRs can provide an efficient and effective means to generate a FSM for each operating mode. Such a FSM is referred as Mode Dependent Fault Signature Matrix (MD-FSM).

Deployment of quantitative residuals for fault diagnosis is also applicable for hybrid systems. For simplicity, a binary coherence vector $C^i = [c_1^i \dots c_m^i]$ is defined with each component c_l^i of C being a Boolean variable whose value is obtained by the following rule, similar to equation (2):

$$c_l^i = \begin{cases} 1 & \text{if } |r_l| > \varepsilon_l^i \\ 0 & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \dots, m \quad (2.26)$$

where i denotes the instantaneous operating mode.

2.2.3 An Illustrative Example

Figure 2.37: A two-tank system

A two-tank system is considered, as given in Figure 2.37. The two-tank system consists of two tanks, regulated centrifugal pump modeled as a source of pressure $p_m(t)$ [Pa] and four valves represented by R_1 , R_2 , R_3 and R_4 . A_1 [m²] and A_2 [m²] are cross-section areas of the two tanks. The system is equipped with two pressure sensors ($p_1(t)$ and $p_2(t)$) to measure the pressure at the bottom of tank A_1 and tank A_2 , respectively; this pressure is proportional to the liquid level, according to:

$$p_i(t) = \rho g h_i(t), \quad i = 1, 2 \quad (2.27)$$

where

- ρ liquid density, [kg/m³];
- g gravity, [m/s²];
- $h_i(t)$ liquid height in the tank, [m].

Each valve has two discrete states ON and OFF. A negligible switching-time between open and closed states is assumed. The valves' dynamics is given by

$$f_j(t) = 0, \quad j = 1, 2, 3, 4 \quad \text{when the valve is closed} \quad (2.28)$$

$$f_j(t) = \frac{\text{sign}(\Delta p(t)) \cdot \sqrt{|\Delta p(t)|}}{R_j} = C d_j \cdot \text{sign}(\Delta p(t)) \cdot \sqrt{|\Delta p(t)|}, \quad j = 1, 2, 3, 4$$

when the valve is open (2.29)

where

- $f_j(t)$ liquid-flow through the valve, [m³/s];
- $\Delta p(t)$ pressure difference across the valve, [Pa].
- $C d_j$ coefficient of discharge, [$\sqrt{\text{kg} \cdot \text{m}}$].

The HBG of the two-tank plant is presented in Figure 2.38. In this bond graph, the flow variable is the liquid volumetric flow (i.e. $[m^3/s]$), and the effort variable is pressure (i.e. $[Pa]$). The two tanks are modeled by the two storage components with coefficient $C_i = A_i / g$, $i = 1, 2$. Each one of the four valves is modeled by a set of resistor with parameter $R_j = 1 / Cd_j$, $j = 1, 2, 3, 4$ and controlled-junction with boolean variable a_j , $j = 1, 2, 3, 4$.

Figure 2.38: The two-tank plant acaused HBG

Figure 2.39: The two-tank plant DHBG

The HBG in Figure 2.38 is acaused. In order to apply the SCAPH algorithm, all controlled-junctions are assigned with their preferred causalities. The procedure of applying SCAPH algorithm with allowing inversion of sensor causalities when necessary is listed as followed:

- 1) Choose controlled junction 1_3 that has no source adjacently connected to it. Assign the junction with its preferred causality where the output variable e_6 of the

junction 1_3 is the input of the 1-port component R_3 .

- 2) Similarly, choose controlled junctions 1_4 and 1_6 , the output variables of these controlled junctions are assigned as inputs of the 1-port components (R_2 and R_4 , respectively).
- 3) Choose controlled junction 1_1 that has a source Se connected to it. This pair indicates that the output of the controlled junction must not be an input variable of normal junction 0_2 . Hence the output of the controlled junction 1_1 is assigned as the input variable of 1-port component R_1 .
- 4) Since all controlled junctions have been assigned, all controlled junctions are treated as normal junctions $\{0, 1\}$ in the following procedures.
- 5) The source component Se is assigned and the causal implication is extended to bond 3.
- 6) Since no other source, proceed to Step7).
- 7) The two storage components C_1 and C_2 are assigned with preferred derivative causality and the sensor causality is inverted. Now, the causal implication using the components constraints to bonds 7, 5, 9 and 11 to complete the SCAPH algorithm with allowing inversion of sensor causalities when necessary.

The DHBG is shown in Figure 2.39. The next step is to derive GARRs from the DHBG model, and only junctions with attached sensor are considered. First the sensor p_1 is considered; the constitutive relation of junction 0_2 is given as

$$f_3 - f_4 - f_5 - f_7 = 0 \quad (2.30)$$

Back tracking bond graph causal-paths from unknown to known variables, leads to

$$f_3 = a_1 f_2 = a_1 \frac{\text{sign}(e_1 - e_3) \cdot \sqrt{|e_1 - e_3|}}{R_1} = a_1 \cdot Cd_1 \cdot \text{sign}(p_{in} - p_1) \cdot \sqrt{|p_{in} - p_1|} \quad (2.31)$$

$$f_4 = C_1 \frac{d}{dt} p_1 = \frac{A_1}{g} \cdot \frac{d}{dt} p_1 \quad (2.32)$$

$$f_5 = a_3 f_6 = a_3 \frac{\text{sign}(p_1) \cdot \sqrt{|p_1|}}{R_3} = a_3 \cdot Cd_3 \cdot \sqrt{p_1} \quad (2.33)$$

$$f_7 = a_2 f_8 = a_2 \frac{\text{sign}(p_1 - p_2) \cdot \sqrt{|p_1 - p_2|}}{R_2} = a_2 \cdot Cd_2 \cdot \text{sign}(p_1 - p_2) \cdot \sqrt{|p_1 - p_2|} \quad (2.34)$$

Substituting (2.31) ~ (2.34) into (2.30), and $GARR_1$ is achieved

$$\begin{aligned} GARR_1 = & a_1 \cdot Cd_1 \cdot \text{sign}(p_{in} - p_1) \cdot \sqrt{|p_{in} - p_1|} - \frac{A_1}{g} \cdot \frac{d}{dt} p_1 - a_3 \cdot Cd_3 \cdot \sqrt{p_1} \\ & - a_2 \cdot Cd_2 \cdot \text{sign}(p_1 - p_2) \cdot \sqrt{|p_1 - p_2|} = 0 \end{aligned} \quad (2.35)$$

Similarly, using the junction 0_5 attached to p_2 yields

$$GARR_2 = a_2 \cdot Cd_2 \cdot \text{sign}(p_1 - p_2) \cdot \sqrt{|p_1 - p_2|} - \frac{A_2}{g} \cdot \frac{d}{dt} p_2 - a_4 \cdot Cd_4 \cdot \sqrt{p_2} = 0 \quad (2.36)$$

2.3 Computational Intelligence

The integration of computational intelligence techniques with BG model-based diagnosis and prognosis is believed to provide solutions to health monitoring of hybrid systems because the computational intelligence techniques make the monitoring process more flexible and simple. For example, if the mode information of hybrid system is unavailable during fault parameter identification, the discontinuous function representing the mode switching might be problematic if analytical methods, such as least square method and Gauss-Newton method, are used for the switching time identification considering the non-differentiable function. Therefore, gradient-free stochastic optimization algorithms, as a branch of computational intelligence, could be efficient alternatives.

Computational intelligence is an offshoot of artificial intelligence. As an alternative to classical artificial intelligence it rather relies on heuristic algorithms such as fuzzy systems, neural networks and evolutionary computation. In addition, computational intelligence also embraces techniques that use swarm intelligence, fractals and chaos theory, artificial immune systems, wavelets, etc. Computational intelligence combines elements of learning, adaptation, evolution and fuzzy logic (fuzzy sets) to create programs that are, in some sense, intelligent. Computational intelligence research does not reject statistical methods, but often gives a complementary view (as is the case with fuzzy systems). Artificial neural network is a branch of computational intelligence that is closely related to machine learning.

2.3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO), developed by Kennedy and Eberhart [97], is a population based stochastic search method inspired by social behavior metaphor.

Unlike genetic algorithm [99, 100], which relies on Darwin's theory of nature selection and the competition between individual chromosomes, the swarm intelligence in the nature is modeled by fundamental Newtonian mechanics in PSO for optimization purposes. This corporative scheme manifests PSO the concise formulation, the ease in implementation and many distinct features in different types of optimizations.

The individuals in PSO are called particles, and each particle has its own position and velocity. The possible solutions to the optimization task can be represented by the particles' positions. During each generation, each particle moves according to its own best solution obtained so far, denoted as $P_{best,r}$, as well as the global best solution obtained by all particles of the swarm so far, denoted as G_{best} . If the problem space is D-dimensional and the swarm size is s , the PSO algorithm can be represented as follows:

$$v_{r,g}^{k+1} = w_r v_{r,g}^k + \Psi_1 rand_1(p_{best,r}^k - x_{r,g}^k) + \Psi_2 rand_2(G_{best}^k - x_{r,g}^k) \quad (2.37)$$

$$x_{r,g}^{k+1} = x_{r,g}^k + v_{r,g}^{k+1} \quad (2.38)$$

where

r	particle index in a swarm, $r = 1, 2, \dots, s$;
g	dimension index in a particle, $g = 1, 2, \dots, D$;
k	iteration number;
w_r	inertia weight of particle r ;
$rand_1, rand_2$	two random values with range $[0, 1]$;
Ψ_1, Ψ_2	acceleration coefficients;
$v_{r,g}^k$	velocity of particle r at iteration k , $-V_{\max} \leq v_{r,g}^k \leq V_{\max}$;
$x_{r,g}^k$	position of particle r at iteration k , $-x_{\max} \leq x_{r,g}^k \leq x_{\max}$;
$P_{best,r}^k$	$P_{best,r}$ at iteration k ;
G_{best}^k	G_{best} at iteration k .

Figure 2.40: Movement of a particle

The velocity of each particle $v_{r,g}^k$ is clamped to the range $[-V_{\max}, V_{\max}]$ to reduce the likelihood of particles leaving search space. Note that this does not restrict the position of each particle $x_{r,g}^k$ to the range $[-V_{\max}, V_{\max}]$; it only limits the maximum distance that a particle will move during one iteration. The inertia weight w_r is similar to the momentum term in a gradient descent neural network training algorithm. Acceleration coefficients Ψ_1 and Ψ_2 also control how far a particle will move in a single iteration. Typically, they are both set to value of 2.0. The movement pattern of a particle is illustrated in Figure 2.40. The PSO algorithm can be summarized as follows:

- 1) Initialize the particle swarm. Initialize s particles with random positions and velocities within the search space.
- 2) Calculate the fitness value of each particle.
- 3) Compare the fitness values at the present position with its best position $P_{best,r}^k$ so far for each particle. If the fitness value at the present position is better than that at its best position, set the present position as the best position so far for this particle. Otherwise, keep the best position unchanged for this particle.
- 4) Compare the fitness value of each particle with the fitness value at the global best position G_{best} within the population. If the fitness value of a particle is better than that at the global best position, set the present position of this particle as the best global position so far for the population. Otherwise, keep the original best global

position for the population.

- 5) Update the velocity and position of each particle by using (2.37) and (2.38).
- 6) Check the termination criterion. If the criterion is met, then stop. Otherwise, go to step (2).

The PSO mentioned above is Real-valued PSO (RPSO) which is developed to solve the problem where solution elements are continuous real number. Many optimization problems are set in a space featuring discrete, qualitative distinctions between variables and between levels of variables. Some examples like ordering or arranging of discrete elements, as in scheduling and routing problems. Other problems need cast floating-point problems in binary terms, and solve them in a discrete number space. As any problem, discrete or continuous, can be expressed in a binary notation, it is seen that an optimization algorithm which operates on two-valued functions could be advantageous.

A type of Binary-valued PSO (BPSO) was introduced to allow the PSO algorithm to operate in binary problem spaces [98]. The BPSO has a structure almost identical to the standard PSO, where the velocity is still defined to be in the continuous space. In abinary space, a particle moves in a state space restricted to zero and one on each dimension. The logistic function of the particle velocity is used as the probability distribution for the position, that is, the particle position in a dimension is randomly generated using that distribution. The equation that updates the particle position is changed as follows:

$$x_{r,g}^{k+1} = \begin{cases} 0 & \text{if } rand \geq f(v_{r,g}^k) \\ 1 & \text{if } rand < f(v_{r,g}^k) \end{cases} \quad (2.39)$$

with

$$f(v_{r,g}^k) = \frac{1}{1 + \exp(-v_{r,g}^k)} \quad (2.40)$$

where *rand* is a random value with range [0,1].

The BPSO is susceptible to sigmoid function saturation, which occurs when velocity values are either too large or too small. In such cases the probability of a change in bit value approaches zero, thereby limiting exploration. For a velocity of 0, the sigmoid function returns a probability of 0.5, implying that there is a 50% chance for the bit to flip.

In order to ensure that there is always some chance of a bit flipping, the velocity can be limited in the range $[-V_{\max} \quad V_{\max}]$ to avoid saturation of the sigmoid function in (2.40). In the BPSO, $V_{\max}$ functions similarly to the mutation rate in GA, smaller $V_{\max}$ allows a higher mutation rate and vice versa. For example, if $V_{\max} = 6.0$, then the probabilities will be limited to $f(v_{r,g}^k)$, between 0.9975 and 0.0025. The result of this is that new vectors will still be tried, even after each bit has attained its best position. Setting $V_{\max}$ with higher value, e.g. 10.0, makes new vectors less likely. Note that while high $V_{\max}$ in the RPSO increases the range explored by a particle, the opposite occurs in the binary version; smaller $V_{\max}$ allows a higher mutation rate. It is also observed that the probability that a bit will be a one is equal to $f(v_{r,g}^k)$, and the probability that it will be a zero is $1 - f(v_{r,g}^k)$. Moreover, if it is a zero already, then the probability that it will change is $f(v_{r,g}^k)$, and if it is a one the probability it will change is $1 - f(v_{r,g}^k)$. Consequently, the probability of the bit changing is given by

$$P_{\text{change}} = f(v_{r,g}^k)[1 - f(v_{r,g}^k)] = f(v_{r,g}^k) - f^2(v_{r,g}^k) \quad (2.41)$$

(2.41) is the absolute (non-directional) rate of change for that bit given a value of $v_{r,g}^k$. Thus change in $v_{r,g}^k$ is still a change in the rate of change, with a new interpretation.

2.3.2 Differential Evolution

Evolutionary Algorithms (EAs) are a class of stochastic search methods that include GA, evolutionary programming, evolution strategies, genetic programming, and their variants. One of the most promising EAs for solving global optimization problems is Differential Evolution (DE) algorithm [110, 111]. DE is proposed at the same year as PSO. DE is a relatively new algorithm among EAs and is capable of dealing with non-differentiable, non-linear, and multimodal objective functions. Many studies showed that DE converges fast and is robust, and simple in implementation. Besides, DE has less parameter which means less dependence on parameter settings [112-114].

DE is a population-based optimization algorithm. The invention of the DE

algorithm goes back to genetic annealing [115]. The genetic annealing is developed to solve the Chebyshev polynomial fitting problem. In order to solve Chebyshev problem in continuous space, the genetic annealing algorithm is modified from bit-string to floating-point encoding and consequently switched from logical operators to arithmetic ones. During experiments, it is discovered that the differential mutation can be used to perturb the population of vectors. It is also noted that by using differential mutation, discrete recombination, and pair-wise selection, there is no need to apply annealing mechanism; it was removed completely and DE was born. Results of this achievement were reported in the ICIS technical report [110]. By this way, the capacity and advantages of DE were introduced to the optimization community. Comprehensive history and development of DE is presented in [116]. The main procedures of DE are described as follows

1) Initialization

Assume the solution space is D-dimensional and the i th individual in DE is represented by a vector $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. In the initialization phase, individuals are randomly generated

$$X_{ij}^0 = X_j^L + rand_j \cdot (X_j^U - X_j^L) \quad (2.42)$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, D$$

where X_j^U and X_j^L are the upper bound and lower bound of the solution with $X_j^L < X_j < X_j^U$, and $rand_j$ is the j th norm random number with $rand_j \in [0 \ 1]$.

2) Mutation

The objective of mutation is to enable search diversity in the parameter space as well as to direct the existing object vectors with suitable amount of parameter variation in a way which will lead to better results at a suitable time. It keeps the search robust and explores new areas in the search domain.

During mutation at generation G , three individuals $X_{r_1}^G$, $X_{r_2}^G$ and $X_{r_3}^G$ satisfying $r_1 \neq r_2 \neq r_3$ and $r_1, r_2, r_3 \in [1, N]$ in population are randomly selected. Using strategy *rand/1/bin*, the mutation operator can be represented as

$$V_{ij}^{G+1} = X_{r_3j}^G + F \cdot (X_{r_1j}^G - X_{r_2j}^G) \quad (2.43)$$

where F is the differential factor satisfying $F \in [0, 2]$.

Figure 2.41: Two dimensional example of an objective function showing its contours and the operation of generating mutant vector $F \cdot (X_{r_1j}^G - X_{r_2j}^G)$ using strategy

DE/rand/1/bin

A two dimensional example of generating mutation vector using strategy *DE/rand/1/bin* is depicted in Figure 2.41. The differential factor F is used to control the amplification of the differential variation $X_{r_1j}^G - X_{r_2j}^G$. Larger values for F result in higher diversity in the generated population and lower values cause faster convergence. It should be noted that new values for r_1 , r_2 and r_3 have to be generated for each value of i , i.e. for each individual in the population.

3) Crossover

DE utilizes the crossover operation to generate new solutions by shuffling

competing vectors and also to increase the diversity of the population. For the classical DE (DE/*rand*/1/*bin*), the binary crossover (shown by “*bin*” in the notation) is utilized. In crossover,

$$U_{ij}^{G+1} = \begin{cases} V_{ij}^{G+1}, & \text{if } rand_j \leq C_r \quad \text{or} \quad j = rnbr(i) \\ X_{ij}^G, & \text{if } rand_j > C_r \quad \text{or} \quad j \neq rnbr(i) \end{cases} \quad (2.44)$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, D$$

where $C_r \in [0,1]$ is the crossover factor, and $rnbr(i)$ is a randomly selected index from D dimensions to ensure that at least one dimension parameter from V_i^{G+1} can be obtained by U_i^{G+1} . Figure 2.42 presents a pictorial example for the binary crossover.

Figure 2.42: A pictorial example for the binary crossover in DE ($rnbr(i)=k=7$)

There are other DE variants which are indicated by the notation DE/*a*/*b*/*c* [117]. In this notation, ‘*a*’ specifies the vector which should be mutated; it can be the best vector (‘best’) of the current population or a randomly selected one (‘rand’). ‘*b*’ is reserved for the number of difference vectors which participate in the mutation (1 or 2) and ‘*c*’ denotes the applied crossover scheme, binary (‘bin’) or exponential (‘exp’). Some examples like DE/ *rand*/1/*bin*, DE/*best*/1/*bin*, DE/ *rand*/2/*bin* and DE/*best*/2/*bin*.

Figure 2.43 presents a pictorial example for the exponential crossover. Crossover procedure starts from a randomly selected parameter (3 in this example), and copies the parameters of the V_{ij}^{G+1} to U_{ij}^{G+1} , until the first occurrence of $rand(0, 1) > C_r$, then the remaining parameters from X_{ij}^G are inherited by U_{ij}^{G+1} .

Figure 2.43: A pictorial example for the exponential crossover in DE

4) Selection

After mutation and crossover, check whether the individuals generated are still in the search range $[X_j^L, X_j^U]$, and if any parameter is out of the range, this parameter is randomly regenerated within the range,

$$U_{ij}^0 = U_j^L + rand_j \cdot (X_j^U - X_j^L), \quad i = 1, 2, \dots, N; j = 1, 2, \dots, D \quad (2.45)$$

Selection is an approach which must decide which vector (U_{ij}^{G+1} or X_{ij}^G) should be a member of next (new) generation, $G+1$. For a minimization problem, the vector with the lower value of objective function is chosen (greedy selection).

Selection operates by comparing the individuals' fitness to generate the new population of next generation,

$$X_i^{G+1} = \begin{cases} U_i^{G+1}, & \text{if } f(U_i^{G+1}) > f(X_i^G) \\ X_i^G, & \text{otherwise} \end{cases} \quad (2.46)$$

DE was initially designed for global optimization problems over continuous spaces. The standard DE algorithms cannot be applied as is to solve problems with binary-valued parameters. Another version of DE, called Binary DE (BDE) is proposed, in which the solution is represented by a binary string instead of a real value vector [19]. Since crossover operates as discrete form in continuous space, it can be directly used in BDE. Mutation is the only operator which should be modified.

$$D_j(\text{Mutant}, X_{r_1}^G) = \begin{cases} 1, & \text{if } \text{rand}_j < F \wedge D_j(X_{r_2}^G, X_{r_3}^G) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.47)$$

In (2.47), for each position j , when a randomly selected number less than F and the *hamming distance* $D_j(X_{r_2}^G, X_{r_3}^G)$ is equal to 1, then decision variable $D_j(\text{Mutant}, X_{r_1}^G)$ is equal to 1 which means that the parameter at position j can be mutated. This type of mutation is called restricted-change DE mutation.

2.3.3 Fuzzy Logic System

Fuzzy set and theory was first proposed by Zadeh of the University of California at Berkeley in a 1965 [162]. The fuzzy logic system is the embodiment of fuzzy sets and logic theory in engineering. It enables the incorporation of information from human experts who describe their knowledge about the system in natural language to the system design. Fuzzy logic is widely used in control system design. Although genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

Fuzzy set theory is an alternative to crisp set theory. In crisp set theory, an element is either in a given set, or it is not in this set. Fuzzy set theory declares that all elements are in the given set to a certain degree. The degree to which an element is in the set (i.e. the element's degree of membership in the set) is defined by the membership function of the set. The degree of membership for an element is generally set to be in the range $[0, 1]$. An important feature of fuzzy system theory is that it provides a systematic procedure for transforming a knowledge base into a nonlinear mapping. If fuzzy logic system reflects the human expert knowledge about control experience, then fuzzy logic system functions as a fuzzy logic controller. If fuzzy logic system incorporates expert's knowledge related to the dynamic characteristics of the plant in the fuzzy rules, then the fuzzy logic system can be considered as a fuzzy model. While a fuzzy logic system acts as the parameters tuning mechanism for controller, then the fuzzy logic system is a fuzzy tuner of controller parameters. A fuzzy logic system usually consists of three modules and a general scheme of fuzzy

logic system is depicted in Figure 2.44.

Figure 2.44: Basic configuration of a fuzzy logic system

1) Fuzzification

It is a process of mapping from the real number of input variables to fuzzy sets described in linguistic variables. Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable is generally decomposed into a set of linguistic terms, such as cold, warm, and hot for linguistic variable temperature. Membership functions are used in the fuzzification and defuzzification steps of a fuzzy logic system, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function is used to quantify a linguistic term.

For example, a fuzzy set A of a universe of discourse X is characterized by a membership function $\mu_A(x)$, which assigns to each element $x \in X$ a number $\mu_A(x)$ between 0 and 1, which represents the degree of membership in A . A typical set of triangular membership functions are given in Figure 2.45 with a universe of discourse $[0, 20]$. The real value input $x=8$ belongs, with a membership degree $\mu_{A_1}(8) = 0.5$, to the fuzzy set A_1 and the membership degree $\mu_{A_2}(8) = 0.5$, to the fuzzy set A_2 . The fuzzy sets are described by linguistic variables, such as VS (Very Small), MS (Medium Small), MB (Medium Big) and VB (Very Big). Note that, an important characteristic of fuzzy logic is that a numerical value does not have to be fuzzified using only one membership function. In other words, a real value can belong to

multiple sets at the same time. There are different shapes of membership function such as triangular, trapezoidal, bell-shaped, Gaussian, or singleton.

Figure 2.45: Fuzzification process based on triangular membership functions

2) Fuzzy inference engine with a rule base

Once the membership functions have been defined for the input and output variables, a fuzzy rule must be designed. Fuzzy rule determines what output takes place under what input conditions. In general, fuzzy rule is described on the basis of the linguistic IF-THEN form

IF (conditions) THEN (consequences).

The IF part of the conditional statement is called the antecedent of a rule, while the THEN part is called the consequent of a rule. Usually the antecedent and the consequent of a rule are associated with several variables (input variables and output variables). If one input and one output is used, the rules can be represented by

$$\begin{aligned}
 R_1 &: \text{IF } x \text{ is } A_1 \text{ THEN } y \text{ is } B_1 \\
 R_2 &: \text{IF } x \text{ is } A_2 \text{ THEN } y \text{ is } B_2 \\
 &\vdots \\
 R_i &: \text{IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i \\
 R_n &: \text{IF } x \text{ is } A_n \text{ THEN } y \text{ is } B_n
 \end{aligned} \tag{2.48}$$

where x and y are input and output linguistic variables of the fuzzy system with $x \in X$ and $y \in Y$, X and Y are input and output space (universe of discourse); A_i and B_i are input and output fuzzy sets, respectively, and n is the number of rules in the fuzzy system, i.e. $i = 1, 2, \dots, n$.

Each fuzzy rule is represented a fuzzy implication R_i from fuzzy input sets to fuzzy output sets, denoted by $A_i \rightarrow B_i$, and its membership is defined as follows

$$\mu_{R_i}(x, y) = \mu_{A_i \rightarrow B_i}(x, y) = \min(\mu_{A_i}(x), \mu_{B_i}(y)) \quad (2.49)$$

where $\min$ is a minimum function. Equation (2.49) shows the minimum rule of fuzzy implication among several kinds of implication rules. If a fuzzy input A_i and the fuzzy relation R_i is known, the corresponding fuzzy output can be determined by taking the composition of A_i and R_i . Using the max-min composition rule of inference, proposed by Mamdani [163], the output from a rule R_i can be represented as

$$\mu_{B_i}(y) = \max_{x \in X} \left[\min(\mu_{A_i}(x), \mu_{R_i}(x, y)) \right] \quad (2.50)$$

where $\max$ is a maximum function. The overall fuzzy output inferred from the set of rules is obtained by combining the sets of fuzzy relations $\{R_i\}$ by union

$$\mu_B(y) = \max_{i=1}^n \left\{ \max_{x \in X} \left[\min(\mu_{A_i}(x), \mu_{R_i}(x, y)) \right] \right\} \quad (2.51)$$

3) Defuzzification

The fuzzy control output B from the inference engine is a fuzzy value. In order to implement the method in a real system, real value output is needed. The defuzzification process converts the output fuzzy sets into real number of output variable. There are several defuzzification methods, and the center of gravity method is the most commonly used method and can be described as

$$y^* = \frac{\sum_{i=1}^n \bar{y}_i \mu_{B_i}(\bar{y}_i)}{\sum_{i=1}^n \mu_{B_i}(\bar{y}_i)} \quad (2.52)$$

where $\bar{y}_i = \arg \max [\mu_{B_i}(y)]$, i.e. $\bar{y}_i$ is the point in Y at which $\mu_{B_i}(y)$ attains its maximum value.

Chapter 3

Simultaneous Fault Parameter and Mode Change Identification

When mode switching events of a hybrid system are exactly known, FDI of such systems is easy because the knowledge of the active regime at any moment is available. However, lack of mode switching information will make it difficult to carry out a fault diagnosis scheme. This chapter addresses the determination of mode switching and estimation of fault parameters for hybrid systems using Adaptive Hybrid Particle Swarm Optimization (AHPSO). A novel individual level adaptive mechanism using fuzzy system is proposed to dynamically adjust the parameters of the algorithm. GARRs are used to describe the behavior of a hybrid system at all modes. This method takes advantages of both AHPSO and GARRs to simultaneously identify fault parameters and mode switching in a hybrid system. Two illustrative examples, i.e. an electro-hydraulic suspension and an electrical circuit, are studied using the proposed algorithm.

3.1 Quantitative Hybrid Bond Graph Diagnosis

Figure 3.1 presents the architecture of the QHBG based quantitative diagnosis framework [39, 96]. The framework utilizes the unified information of the HBG to monitor a hybrid system efficiently and effectively. This framework consists of a GARR alarm generator, a fault detection module, a fault isolation module, a fault estimator, and a mode tracker. These modules are based on the HBG of the hybrid system. The mode tracker constantly determines the instantaneous operating mode using sensors and input information that are obtained from the hybrid system. This

Figure 3.1: Architecture of quantitative diagnosis framework

instantaneous mode information allows users to evaluate the GARRs for residuals effectively at all operating modes. With these residuals, users are able to detect, isolate, and estimate the size of a fault. The fault detection module decides whether or not a fault has occurred when any of the residual signals is non-zero. Here assume that the system can have only one fault in order to uniquely isolate the fault parameter. For those fault isolable parameters, the GARRs generate a set of unique residuals that allows the faulty parameter to be identified. For those non-isolable but detectable parameters, the module will select a set of potential fault candidates. Finally, the parameter estimation module estimates the selected fault parameters to access the health status of the system. This information can also be used to refine the non-isolable parameter.

A. Fault Detection and Isolation Modules

If the hybrid system is fault-free, then the binary coherence vector C^i defined in (2.26) will be a zero vector, where i denotes the instantaneous operating mode. On the other hand, if the system is faulty, then the vector C^i will be a nonzero vector. If a fault

occurs at time t , in order to isolate the faulty parameter, the binary coherence vector C^i is matched with the fault signature of the FSM of the operating mode at time t . Assume that the system will have a single fault, so the fault can be detected and uniquely isolated based on its fault detectability and fault isolability.

B. Mode Tracker

The purpose of mode tracker is to determine the mode information of the monitored hybrid system [42, 43]. In HBG, system modes are determined by the controlled junction states. The formulas of GARRs include controlled junction state variables. This means that the structure properties of GARRs can be explored to deduce the relationship between GARR response and mode changes. Evaluation of GARRs produces residuals. The structure of the residuals form a binary matrix M_{ij} whose value is taken as

$$M_{ij} = \begin{cases} 1 & \text{if the constitutive equation of the controlled } j\text{th junction is described in the } i\text{th residual} \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

The matrix which represents cause-effect relations between mode changes and GARRs is called the Mode Change Signature Matrix (MCSM). MCSM provides the logic for the mode-change isolation after an inconsistency is detected. Each controlled junction has a corresponding signature and its mode-change is isolable if its signature is unique in the matrix. Similar to FSM based continuous system FDI, mode-change detectability (D_b) and mode-change isolability (I_b) are listed at the last two columns of the MCSM.

This method assumes that any change of mode is due to a change of a single state variable (this assumption is analogous to the single fault assumption, in the context of parametric-fault isolation). Using this assumption, any row of the MCSM is a mode change signature and all possible mode-changes are represented by the matrix (some of the signatures may be null). The mode change signatures represent cause-effect relations between mode-changes and GARRs.

Since mode tracking is not required during system normal operation, mode tracker consists of two phases: full ARR-based mode identification and narrowed ARR-based

mode identification. Full ARR-based mode identification is a process that is based on the complete set of ARRs (of all modes), while narrowed ARR-based mode identification is a process that is based on a subset of ARRs due to the MCSM based mode change isolation. Usually, if the hybrid system initial mode is unknown, the first step is to use full ARR-based mode identification to estimate the initial mode. If an inconsistency is detected, a coherence vector $C^i = [c_1^i \ c_2^i \ \dots \ c_m^i]$ is generated, where i denotes the instantaneous operating mode and c_m^i is a binary variable representing the consistency of GARR_m, i.e. $c_m^i = 0$ means GARR_m is consistent with monitored system; otherwise $c_m^i = 1$. Then the coherence vector is compared with the MCSM to generate the mode hypotheses. In order to identify the mode, the sets of ARRs at these suspected modes are run simultaneously. If a new mode is identified, then it is substituted into the GARRs and the role of the mode tracker is ended until next inconsistency is detected. If a new mode is not identified by the mode tracker, then the inconsistency is not caused by a single mode change. A fault hypothesis set is generated for further refinement.

C. Fault Parameter Estimator

The parameter estimation module estimates the selected fault parameters to assess the health status of the system. This information can also be used to refine the non-isolable parameter. The GARRs is exploited to formulate the parameter estimation problem as a nonlinear least-square problem, and solves it using Gauss-Newton method.

From parameter identification viewpoint, $g_l(\theta, n) = 0$ if the parameters θ are the physical parameters of the system. $g_l(\theta, n)$ represents $g_l(\theta, \alpha, De, Df, u)$ in (2.24) for ease of illustration because all other variables are known and are not concerned by the identification algorithm, and n denotes sample index. Generally, $g_l(\theta, n)$ is a nonlinear function of θ , so without loss of generality, the estimation process is formulated as a nonlinear least-square problem. The objective function for the nonlinear least square problem based on N samples of data as follows

$$J(\theta) = \frac{1}{2} G^T G \quad (3.2)$$

where $G = [G_1 \ G_2 \ \cdots \ G_N]^T$,

$G_n = [g_1(\theta, n) \ g_2(\theta, n) \ \cdots \ g_m(\theta, n)]^T$ for $n=1,2,\dots,N$, N is the totally number of samples.

According to Gauss-Newton method, the iterative solution which minimizes the objective function (3.2) is

$$\theta_{k+1} = \theta_k - H_J(\theta_k)^{-1} \nabla J(\theta_k) \quad (3.3)$$

where k denotes the iteration index,

$$\nabla J(\theta) = \frac{\partial G(\theta_k)^T}{\partial \theta} \cdot G(\theta_k),$$

$$H_J(\theta_k) = \frac{\partial G(\theta_k)^T}{\partial \theta} \cdot \frac{\partial G(\theta_k)}{\partial \theta}.$$

This fault estimation algorithm also applies to the case when a parameter of a system's fault is nonlinear parameterized. Another advantage of this method is that it allows mode changes happen during the identification process. Unlike the method in [34, 35], when data is collected for fault parameter identification, the mode is assumed to be persistent.

3.2 Parameterization of Mode Changes

Isolation of fault candidates leads to the identification of fault parameters and evaluation of system health. However, if mode switching is unknown, the choice of a correct model is difficult after fault occurrence, model based mode tracker cannot function well. Nonlinear least-square based parameter estimation module in the QHBG framework requires mode information for identification purpose. In order to overcome this problem, the mode changes are parameterized such that they can be represented by the functions of switching time stamps. During the estimation process, the fault parameters together with the switching time stamps are identified by the estimation algorithm to evaluate the size of the fault parameter.

In this chapter, a fault candidate is represented by a component parameter. It is also assumed that there are at most of p mode changes during an observation period.

Definition 1: For a hybrid system, during the observation window $[t_0 \ t_f]$,

$\alpha = [a_1, a_2, \dots, a_q]^T$ is defined as a switching vector whose component a_j , $j = 1, 2, \dots, q$, takes the form of

$$a_j(t, T_j) = \lambda_j \cdot a_j^1(t, T_j) + (1 - \lambda_j) \cdot a_j^0(t, T_j) \quad (3.4)$$

where λ_j are the Initial Mode Coefficients (IMC), whose values are obtained by the following formula

$$\lambda_j = \begin{cases} 1 & \text{if the state of } j\text{th control junction at time } t_0 \text{ is } 1 \\ 0 & \text{if the state of } j\text{th control junction at time } t_0 \text{ is } 0 \end{cases} \quad (3.5)$$

$a_j^1(t, T_j)$ and $a_j^0(t, T_j)$ are defined as

$$a_j^1(t, T_j) = \tau(t) - \tau(t - T_j^1) + \tau(t - T_j^2) - \dots + (-1)^p \tau(t - T_j^p) \text{ for } \lambda_j = 1 \quad (3.6)$$

$$a_j^0(t, T_j) = \tau(t - T_j^1) - \tau(t - T_j^2) - \dots - (-1)^p \tau(t - T_j^p) \text{ for } \lambda_j = 0 \quad (3.7)$$

Define function

$$\tau(*) = \begin{cases} 1 & \text{if } * \geq 0 \\ 0 & \text{if } * < 0 \end{cases} \quad (3.8)$$

where $T_j = [T_j^1 \ T_j^2 \ \dots \ T_j^p]$ is the switching time stamp vector whose elements are the time stamps for mode switching with $t_0 \leq T_j^1 \leq T_j^2 \leq \dots \leq T_j^p \leq t_f$.

□

By definition, if all the switching time stamps T_j^d , $j = 1, 2, \dots, q$ and $d = 1, 2, \dots, p$ are equal to t_f , then there are no mode changes during the time interval $[t_0 \ t_f]$. If the switching time stamps and IMC are known, the mode switching vector $\alpha = [a_1, a_2, \dots, a_q]^T$ can be expressed as functions of time t .

Figure 3.2: Mode change sequence when the initial state is 1

Figure 3.3: Mode change sequence when the initial state is 0

The essence of definition 1 is the choice of $a_j^1(t, T_j)$ and $a_j^0(t, T_j)$, which can be directly observed from Figure 3.2 and Figure 3.3. It is obvious that the mode changes can be represented as the combination of step functions with different firing time.

3.3 Identification of Fault Parameters and Mode Switching

The formula in equation (3.4) is a discontinuous function, which might be problematic if gradient based methods are used for the switching time identification problem considering the non-differentiable function. Therefore, heuristics-based PSO can be an efficient alternative.

3.3.1 Comparison Between Gradient and Gradient Free Methods

Gradient based optimization algorithms are so far the most commonly used techniques in most existing applications due to their effectiveness in solving convex optimization problems and solid mathematical foundation. Nevertheless, new developments in non classical methods are shifting the focus to confidently adapt heuristic tools to similar or more complex problems. The major differences between heuristic and classical optimization methods can be summarized as follows:

- 1) Most heuristic methods are population based methods that search the solution space by a group of possible solutions. In contrast, gradient based methods use a single path to search for the optimal solution. This difference enhances the chances of locating the near global optima in heuristic methods. It reduces the dependency of successful convergence on the starting search point since most

gradient based methods require a good starting point to ensure successful convergence.

- 2) Heuristic methods introduce randomness into their updating rules, while gradient based methods apply deterministic transition rules. This stochastic nature of the transition rules makes the heuristic methods less likely to get trapped in local optimum points.
- 3) Heuristic methods are general tools that can suit various optimization problems such as nonlinear, discrete, mixed type and constrained problems. This is due to the fact that they are derivative free and require only a fitness function to measure the quality of a given solution. Moreover, they require more fitness evaluation with less computation efforts compared with the gradient based methods.

In short, the choice of which optimization tools one should use is problem dependent. Careful analysis of the nature of the optimization problem and constraints can narrow the search process to select the most suitable technique.

3.3.2 Fault Parameter and Mode Change Estimation

After fault isolation, a set of suspected faults is established. Data is then collected for identification purpose. If there is time difference between FDI and the beginning of data collection, IMC is not equal to the mode that the fault is detected and isolated. Therefore, IMC is also the target of identification task. Since the solution of the identification problem can be split into two parts: real values for targeted fault parameters and switching time stamps and binary values for IMC. The Hybrid PSO (HPSO) algorithm is proposed, where RPSO is used to solve targeted fault parameters and switching time stamps identification problem and binary valued BPSO is utilized to solve IMC identification problem. The RPSO and BPSO evolve simultaneously and are coupled through the common fitness function.

Figure 3.4 depicts the particle structure of HPSO algorithm. In the figure, s is the number of fault candidates. Since the number of IMC, which determine the initial state of the controlled junctions during the observation window, is equal to the number of controlled junctions in HBG, therefore, there are number of q IMC for BPSO. Both solutions from RPSO and BPSO have to be evaluated to produce a

potential solution vector because optimization requires the outcomes from these two parts. A fuzzy system is adopted to adjust the key parameters of the algorithm for better performance.

Target Fault Parameters	Switching Time	IMC
$[Pos_1][Pos_2]...[Pos_s]$	$\{[Pos_1]...[Pos_p]\}_1... \{[Pos_1]...[Pos_p]\}_q$	$[Pos_1][Pos_2]...[Pos_q]$
← RPSO_particle →		← BPSO_particle →

Figure 3.4: Particle structure of HPSO algorithm

In this work, it is assumed that no frequent switching occurs in mode changes, and there exists a constant η such that $|T_j^h - T_j^u| > \eta$ for $\forall T_j^h \in (t_0, t_f)$, $\forall T_j^u \in (t_0, t_f)$, $j = 1, 2, \dots, q$, $h = 1, 2, \dots, p$, $u = 1, 2, \dots, p$ and $h \neq u$. Based on this assumption, a frequent switching avoidance strategy is introduced, in which the switching time stamps estimated have to be compared with each other before each particle position is updated, if any of two switching time stamps satisfy $|T_j^h - T_j^u| \leq \eta$, all the switching time stamps will be reset to the following values

$$T_j^d = \frac{d[(t_f - t_0) - 2\eta]}{p} \quad (3.9)$$

The underlying meaning in the above strategy is that when two switching time stamps are close, all the switching time stamps will be distributed evenly on the search space for next iteration search.

In the undertaken problem, the objective is to search for fault parameter values and switching time stamps with the minimum residual values in certain operating mode. To evaluate the fitness, N samples are collected after a fault is detected. If the parameter values estimated are the true physical parameters and mode switching vector is accurately identified, all GARRs, g_l , for $l = 1, 2, \dots, m$, as defined in equation (2.1), are zero. These lead to the definition of the following fitness function

$$F_{\text{fitness}} = 1 / \left(\sum_{l=1}^m \sum_{n=1}^N |g_l^n| + \varepsilon \right) \quad (3.10)$$

with n represents the sampling time index, and ε is a small positive constant to avoid zero division during the search process. Since the GARRs are used to identify the

targeted fault parameters, only those GARRs containing targeted parameters are selected.

In PSO, it is known that the inertia weight is used to control the influence of the previous velocities on the current velocity, thereby influencing the tradeoff between global and local search abilities of RPSO. Since the maximum permissible velocity affects global exploration indirectly and the inertia weight affects it directly, it will generally be better to control global exploration through inertia weight only. Because of its clear physical meaning and its effectiveness, inertia weight has become a very important part of standard particle swarm optimization. The inertial weight has greatly influence on both global and local search; so many attempts have been tried by using various strategies to make its performance better. Shi and Eberhart [101] proposed a linear decreasing strategy in which

$$w^k = w_{\max} - k(w_{\max} - w_{\min}) / k_{\max} \quad (3.11)$$

where

$w_{\max}$	maximal value of w ;
$w_{\min}$	minimal value of w ;
k	iteration number;
$k_{\max}$	maximal iteration in a run.

In this method, the particle in the beginning of the search could move fast, so that a sufficiently optimal region could be quickly accessed, and the inertial weight changes to be smaller with the iteration of the method later, then the particle slow down to search in the local region. Some other similar techniques are proposed to tune the inertia weight for better performance [102, 103]. However, these algorithms are on the population level, which means that every particle shares the same inertial weight in each iteration. Therefore, the inherent difference of particles' search state is ignored.

As for BPSO, it is clear that using small maximum velocities will make the sigmoid function sensitive to the noise introduced by the random number multiplication. This will result in the bit values easily switching between 0 and 1, in turn increases global search ability. On the contrary, bigger maximum velocity will provide finer search. In this work, a novel individual level adaptive scheme is introduced, in which a fuzzy system is employed to dynamically adjust the inertial weight of RPSO and $V_{\max,r}$ of BPSO according to the improvement of each particle.

The improvement index for r th particle is defined by

$$\zeta_r = \frac{1/F_{\text{fitness}}(p_{\text{best},r}^k)}{1/F_{\text{fitness}}(p_{\text{best},r}^{k-1})} = \frac{F_{\text{fitness}}(p_{\text{best},r}^{k-1})}{F_{\text{fitness}}(p_{\text{best},r}^k)} \quad (3.12)$$

According to equation (3.12), when ζ_r is close to 1, it means that the r th particle has a deficient improvement, then it is desirable to increase its search space, so the inertial weight of RPSO should be increased and $V_{\text{max},r}$ of BPSO should be decreased. On the other hand, when ζ_r is close to 0, it means that the r th particle has a significant improvement, then the area this particle needs to explore should be smaller, so the inertial weight of RPSO should be decreased and $V_{\text{max},r}$ of BPSO should be increased accordingly. Based on the qualitative analysis, a fuzzy system is adopted to dynamically adjust the inertial weight of RPSO and $V_{\text{max},r}$ of BPSO. The improvement index ζ_r is employed as the fuzzy input variable, and the change of the inertial weight Δw_r of RPSO and the change of the maximum velocity $\Delta V_{\text{max},r}$ of BPSO are the fuzzy output variables. To get a balance of global search and local search ability of RPSO, w_r cannot be too large or too small, typical w_r is between 0.4 and 0.9. Similarly, $V_{\text{max},r}$ of BPSO is often set at 4, so that there is always a good chance that a bit will change state. Both positive and negative corrections are needed for the inertial weight of RPSO and $V_{\text{max},r}$ of BPSO. Therefore, a range of -0.1 to +0.1 is chosen for Δw_r and a range of -1 to +1 is selected for $\Delta V_{\text{max},r}$.

The input fuzzy variable is defined in five fuzzy sets: very small (VS), small (SM), medium (ME), large (LA) and very large (VL). Each fuzzy set has its own triangular membership function as shown in Figure 3.5 (a). Two output fuzzy variables consist of five fuzzy sets: negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB). Each fuzzy set has its membership function with singleton shape as shown in Figure 3.5 (b) and (c). The singleton output membership functions are utilized here because of the efficient combination with the weighted average defuzzification process, which is easy for implementation. The fuzzy “IF–THEN” inference rules are established in Table 3.1 based on the aforementioned qualitative analysis. The output Δw_r and $\Delta V_{\text{max},r}$ are derived from the fuzzy inference decision and defuzzification operation.

(a) Improvement index ζ_r (b) Change of inertia weight Δw_r in RPSO (c) Change of maximum velocity $\Delta V_{\max,r}$ in BPSO

Figure 3.5: Fuzzy input and output membership functions

Table 3.1: Rules tables for AHPSO

Antecedent	Consequent	
	Δw_r	$\Delta V_{\max,r}$
VS	NB	PB
SM	NS	PS
ME	ZE	ZE
LA	PS	NS
VL	PB	NB

The block diagram of AHPSO algorithm is shown in Figure 3.6, and it is seen that after the updating of RPSO solutions and BPSO solutions, their combination will be evaluated by the fitness function for next iteration. Moreover, when new candidate solutions are generated in the next iteration, they should be checked before they enter new updating using frequent switching avoidance strategy. Finally, when the number of iteration exceeds the predefined maximal iteration, the algorithm will stop and then output the final solutions which contain fault parameter values, switching time stamps and IMC. The fault parameter values will be compared with their corresponding

nominal values, and those parameters whose values are distinct from their nominal values are considered as true fault parameters.

Figure 3.6: Block diagram of AHPSO algorithm

3.4 Case Study of an Electro-hydraulic Suspension

Figure 3.7: Model of a quarter car

The model of the quarter-car active suspension system, shown in Figure 3.7, is used in this investigation. In the figure, some physical variables are defined as follows:

M_s	sprung mass of quarter car body;
M_u	unsprung mass of wheel;
K_s, K_t	spring coefficients of suspension and wheel, respectively;
x_s, x_u	displacement of the quarter car body and wheel, respectively;
C_s	damper coefficient;
P_1, P_2	pressure inside the two chambers of the cylinder, respectively;
P_s	supply pressure of hydraulic actuator;
P_r	return pressure of hydraulic actuator;
r	road disturbance due to road roughness.

Figure 3.8: Actuator with four way spool valve

Two types of discrete transitions for this system are considered in the study. The first one is autonomous mode change in the four way spool valve. The spool valve flow rate dynamic equation has two modes, depending on the valve position as shown in Figure 3.8, where V_1, V_2 are the original total control volumes of the two cylinder

chambers, respectively (including the volume of the servo valve, pipelines, and cylinder chambers), A_1 , A_2 are the ram area of the two chambers, respectively, x_v is the displacement of spool valve. The mode changes are governed by the values of internal state variables.

Using the equation for hydraulic fluid flow through an orifice, the relationship between spool valve displacement x_v , and the supply flow rate to the forward chamber (or cylinder end) Q_1 , the return flow rate of the return chamber (or rod-end) Q_2 , is given as [104]

$$\begin{aligned} Q_1 &= C_d w x_v \left[s(x_v) \sqrt{\frac{2}{\rho} (P_s - P_1)} + s(-x_v) \sqrt{\frac{2}{\rho} (P_1 - P_r)} \right] \\ Q_2 &= C_d w x_v \left[s(x_v) \sqrt{\frac{2}{\rho} (P_2 - P_r)} + s(-x_v) \sqrt{\frac{2}{\rho} (P_s - P_2)} \right] \end{aligned} \quad (3.13)$$

with

$$s(*) = \begin{cases} 1 & \text{if } * \geq 0 \\ 0 & \text{if } * < 0 \end{cases} \quad (3.14)$$

where

- C_d discharge coefficient;
- ρ fluid density;
- w spool valve area gradient.

The second type of discrete transition is the road profile which has multiple modes [56]. Since the transition from one mode to another mode is not determined by the system internal state, and is also not decided by the driver, this type of mode transition can be considered as exogenous mode changes which are unknown for the designer, the proposed method can be used to identify these mode changes. Note that in hybrid systems, some mode-changes are known, because they are initiated by a supervisory controller which brings different control input at different mode. When consider road profile, it may be characterized as the bumpy road, and the bump will be interpreted as equivalent velocity as required for bond graph flow input. Since during the vehicle travel, it will undergo several different road conditions, like plane road, sandy road and the like. The transition from one road profile to another may be characterized as mode change which will bring different input (excitation), although not controllable,

at different mode. This interpretation is similar to the supervisory controller of hybrid system and assumes that the transition is instantaneous. If the road profile does not match any of the given profiles, the proposed method will not work. As a result, assume that all types of road profiles the vehicle will go through are known in advance.

Without loss of generality, the road profile can be formulated as

$$r = a^T SF \quad (3.15)$$

where

SF road input vector, $SF = [Sf_1 \quad Sf_2 \quad \dots \quad Sf_z]^T$, where z is the total number of road profiles that the vehicle goes through;

a state vector of controlled junctions, $a = [a_1 \quad a_2 \quad \dots \quad a_z]^T$.

It is assumed that the following relationship be satisfied, meaning that at each time moment the vehicle can only go through one road profile

$$a_1 + a_2 + \dots + a_z = 1 \quad (3.16)$$

For simplicity, only two road profiles are considered. One is random road, and the other is bumpy road. The road profile under random road can be represented as [105]

$$\dot{x}_1(t) = 2\pi q_0 \sqrt{G_0 V} w_n(t) \quad (3.17)$$

where

q_0 reference spatial frequency;

G_0 road roughness coefficient;

V vehicle forward velocity;

$w_n(t)$ zero-mean white noise.

The road displacement due to a bump is represented by [106]

$$x_2(t) = \begin{cases} \frac{A}{2} \left(1 - \cos\left(\frac{2\pi V}{L} t\right) \right) & 0 \leq t \leq \frac{L}{V} \\ 0 & t > \frac{L}{V} \end{cases} \quad (3.18)$$

where

A height of the bump;

L length of the bump.

HBG model of electro-hydraulic suspension is developed in Figure 3.10. Here four controlled junctions, i.e., 0_{a1} , 0_{a2} , 1_{a3} and 1_{a4} , are used and Boolean variables a_1 ,

a_2 , a_3 and a_4 are used to represent the state (ON/OFF) of these junctions.

The following relationships are valid

$$\begin{aligned} a_1 &= 1 - a_2 \\ a_3 &= 1 - a_4 \end{aligned} \quad (3.19)$$

where a_3 depends on the state of the displacement of valve x_v

$$a_3 = \begin{cases} 1 & \text{if } x_v \geq 0 \\ 0 & \text{if } x_v < 0 \end{cases} \quad (3.20)$$

Then the road input can be represented as follows

$$r = a_1 \dot{x}_1 + a_2 \dot{x}_2 \quad (3.21)$$

where

$\dot{x}_1$ random road profile;
 $\dot{x}_2$ bumpy road profile.

Note that at junction 1₉, there are two bonds are flow-deciding bonds, i.e., two bonds do not have causal bars at the junction. Traditionally, this is an unacceptable 1-junction model. However, based on equation (3.19), it is clear that the junction flow is chosen from either of the two flow-deciding bonds at mutually exclusive time instants, there should be no conflict with the BG modeling principle. If the flow-deciding bond connected to $\dot{x}_2$ is given in dotted line as shown in Figure 3.9, this clarification can also be specified. In Figure 3.9, $De : P_1$ and $De : P_2$ represent the pressure sensors; $Df : \dot{x}_s$ and $Df : \dot{x}_u$ denote the velocity sensors; $Se : P_s$ represents the supply pressure. Furthermore, the block “Valve” denotes the relationship between control voltage and spool valve

$$\frac{x_v}{u} = \frac{K_v}{\left(\frac{s}{\omega_v}\right)^2 + 2\xi \frac{s}{\omega_v} + 1} \quad (3.22)$$

where

K_v gain of the servo-valve;
 ω_v natural frequency of the servo-valve;
 ξ damping ratio of the servo-valve.

Figure 3.9: HBG of quarter-car electro-hydraulic suspension system

Resistor elements R_1 , R_2 , R_3 and R_4 , are the nonlinear relationships between flow rate and pressure in (3.13). The following nonlinear functions are used to represent the constitutive relations of these elements

$$\begin{aligned}
 f_3 &= g_1(e_3) = -R_1 \sqrt{e_3} = -C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{e_3} \\
 f_6 &= g_2(e_6) = R_2 \sqrt{e_6} = C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{e_6} \\
 f_{13} &= g_4(e_{13}) = -R_3 \sqrt{e_{13}} = -C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{e_{13}} \\
 f_{15} &= g_3(e_{15}) = R_4 \sqrt{e_{15}} = C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{e_{15}}
 \end{aligned} \tag{3.23}$$

From (3.23), it is observed that the coefficient for these resistor elements, i.e. $C_d \cdot w \cdot x_v \sqrt{\frac{2}{\rho}}$, is not constant because it is modulated by the displacement of spool valve x_v , thus R_1 , R_2 , R_3 and R_4 are modulated resistors. In Figure 3.9, these modulated resistors are represented by common R -elements with additional information bonds from the signal x_v .

External leakage flow is negligible in this study. Resistive element $R:1/C_m$ represents the internal leakage of the cylinder. TF_1 and TF_2 represent the relationship between velocity of the single-rod of the cylinder and the flow rate changes of two chambers. C_1 and C_2 represent the fluid compliance of the two chambers of the cylinder. The constitutive relations are nonlinear functions as follows

$$f_{11} = g_{C_1}(e_{11}) = C_1 \frac{d}{dt}(e_{11}) = \frac{V_1 - A_1 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(e_{11}) \quad (3.24)$$

$$f_7 = g_{C_2}(e_7) = C_2 \cdot \frac{d}{dt}(e_7) = \frac{V_2 + A_2 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(e_7) \quad (3.25)$$

where

β effective bulk modulus of the hydraulic fluid.

By inspection of (3.24), it is observed that the parameter representing C -element, i.e. $\frac{V_1 - A_1 \int \dot{x}_s dt}{\beta}$, is not a constant and is modulated by the flow sensor measuring the velocity of the quarter car body $\dot{x}_s$, so it is a modulated C -element. Thus, in the model of Figure 3.9, it is modeled by using standard C -element with additional information bond from the time integral of signal $\dot{x}_s$. The same solution also applies to C_2 .

Elements $C:1/K_t$ and $C:1/K_s$ in Figure 3.9 represent the compliance of two springs; $R:C_s$ denotes the damper coefficient; $I:M_s$ and $I:M_u$ stand for the inertias of the unsprung mass and sprung mass, respectively.

The GARRs can be derived from the Diagnostic Hybrid Bond Graph (DHBG) of the system. DHBG is a HBG with suitable causalities such that every active BG component remains valid at all operating modes. This feature of the DHBG allows

consistent causal description and derivation of GARRs of a hybrid system from its DHBG. By applying the procedure developed in [40], the DHBG of the hybrid system is obtained as shown in Figure 3.10.

Figure 3.10: DHBG of quarter-car electro-hydraulic suspension system

The constitutive relations of junctions 0_1 , 0_2 , 1_3 , 0_4 , TF_1 , TF_2 , 1_5 , 0_6 , 1_7 , 0_8 and 1_9 are given by the following equations

$$a_3 f_2 + a_4 f_{16} - f_1 = 0 \quad (3.26)$$

$$a_3 f_4 + f_8 + f_{17} - f_7 - a_4 f_5 = 0 \quad (3.27)$$

$$e_{10} - e_8 - e_9 = 0 \quad (3.28)$$

$$a_4 f_{14} - f_{10} - f_{11} - a_3 f_{12} - f_{19} = 0 \quad (3.29)$$

$$f_{19} - f_{20}/TF_1 = 0 \quad (3.30)$$

$$e_{20} - e_{19}/TF_1 = 0 \quad (3.31)$$

$$e_{18} - TF_2 e_{17} = 0 \quad (3.32)$$

$$f_{17} - TF_2 f_{18} = 0 \quad (3.33)$$

$$e_{20} + e_{23} - e_{18} - e_{21} - e_{22} = 0 \quad (3.34)$$

$$f_{25} - f_{23} - f_{24} = 0 \quad (3.35)$$

$$e_{27} - e_{25} - e_{26} = 0 \quad (3.36)$$

$$f_{29} - f_{27} - f_{28} = 0 \quad (3.37)$$

$$a_1 e_{32} - a_2 e_{30} - e_{29} = 0 \quad (3.38)$$

Four structurally independent GARR equations can be generated from equations (3.27), (3.29), (3.34) and (3.37) after eliminating the unknown variables. The unknown variables elimination process is achieved by following the causal path, from known to unknown, on the DHBG model. For equation (3.27), the unknown variables f_4 , f_5 , f_7 , f_8 and f_{17} can be calculated as follows

$$f_4 = f_3 = g_1(e_3) = g_1(e_2 - e_4) = -C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_s - P_2} \quad (3.39)$$

$$f_5 = f_6 = g_2(e_6) = g_1(e_5) = C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_2} \quad (3.40)$$

$$f_7 = g_{C_2}(e_7) = \frac{V_2 + A_2 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(P_2) \quad (3.41)$$

$$f_8 = f_9 = C_{m_i} \cdot (P_1 - P_2) \quad (3.42)$$

$$f_{17} = TF_2 \cdot f_{18} = -A_2 \cdot \dot{x}_s \quad (3.43)$$

Then the first GARR is obtained as follows

$$\begin{aligned} GARR_1 = & -a_3 \cdot C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_s - P_2} + C_{m_i} \cdot (P_1 - P_2) - A_2 \cdot \dot{x}_s \\ & - \frac{V_2 + A_2 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(P_2) - a_4 \cdot C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_2} \end{aligned} \quad (3.44)$$

For equation (3.29), the unknown variables f_{10} , f_{11} , f_{12} , f_{14} and f_{19} can be calculated as follows

$$f_{10} = f_9 = C_{im} \cdot (P_1 - P_2) \quad (3.45)$$

$$f_{11} = g_{C_1}(e_{11}) = \frac{V_1 - A_1 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(P_1) \quad (3.46)$$

$$f_{12} = f_{13} = g_4(e_{13}) = g_4(e_{12}) = -C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_1} \quad (3.47)$$

$$f_{14} = f_{15} = g_3(e_{15}) = g_3(e_{16} - e_{14}) = C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_s - P_1} \quad (3.48)$$

$$f_{19} = f_{20}/TF_1 = -A_1 \cdot \dot{x}_s \quad (3.49)$$

Then the second GARR is obtained as follows

$$\begin{aligned} GARR_2 = & a_4 \cdot C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_s - P_1} - C_{im} \cdot (P_1 - P_2) + A_1 \cdot \dot{x}_s \\ & - \frac{V_1 - A_1 \int \dot{x}_s dt}{\beta} \cdot \frac{d}{dt}(P_1) + a_3 \cdot C_d \cdot w \cdot x_v \cdot \sqrt{\frac{2}{\rho}} \cdot \sqrt{P_1} \end{aligned} \quad (3.50)$$

For equation (3.34), the unknown variables e_{18} , e_{20} , e_{22} and e_{23} can be calculated as follows

$$e_{18} = TF_2 e_{17} = -A_2 \cdot P_2 \quad (3.51)$$

$$e_{20} = e_{19}/TF_1 = -A_1 \cdot P_1 \quad (3.52)$$

$$e_{22} = C_s \cdot \dot{x}_s \quad (3.53)$$

$$e_{23} = e_{24} = M_s \cdot \frac{d}{dt}(f_{24}) = M_s \cdot \frac{d}{dt}(\dot{x}_u - \dot{x}_s) \quad (3.54)$$

The constitutive relation of the component $C : 1/K_s$

$$\frac{1}{K_s} \cdot \frac{d}{dt}(e_{21}) = \dot{x}_s \quad (3.55)$$

Then the third GARR is obtained from the component $C : 1/K_s$ after combining (3.51) ~ (3.55)

$$\begin{aligned} GARR_3 = & -A_1 \cdot \frac{1}{K_s} \cdot \frac{d}{dt}(P_1) + M_s \cdot \frac{1}{K_s} \cdot \frac{d^2}{dt^2}(\dot{x}_u - \dot{x}_s) \\ & + A_2 \cdot \frac{1}{K_s} \cdot \frac{d}{dt}(P_2) - C_s \cdot \frac{1}{K_s} \cdot \frac{d}{dt}(\dot{x}_s) - \dot{x}_s \end{aligned} \quad (3.56)$$

For equation (3.37), the unknown variables f_{27} , f_{28} and f_{29} can be calculated as follows

$$f_{27} = \dot{x}_u \quad (3.57)$$

$$f_{28} = \frac{1}{K_t} \cdot \frac{d}{dt}(e_{25} + e_{26}) = \frac{1}{K_t} \cdot \frac{d}{dt} \left[M_s \cdot \frac{d}{dt}(\dot{x}_u - \dot{x}_s) + M_u \cdot \frac{d}{dt}(\dot{x}_u) \right] \quad (3.58)$$

$$f_{29} = a_1 f_{33} + a_2 f_{31} = a_1 \dot{x}_1 + a_2 \dot{x}_2 \quad (3.59)$$

Then the fourth GARR is obtained as follows

$$GARR_4 = a_1 \dot{x}_1 + a_2 \dot{x}_2 - \frac{1}{K_t} \cdot \frac{d}{dt} \left[M_s \cdot \frac{d}{dt}(\dot{x}_u - \dot{x}_s) + M_u \cdot \frac{d}{dt}(\dot{x}_u) \right] - \dot{x}_u \quad (3.60)$$

The following MD-FSM tables are obtained

Table 3.2: MD-FSM at $a_1 = 1$, $a_3 = 1$

	r_1	r_2	r_3	r_4	D_b	I_b
C_d	1	1	0	0	1	0
w	1	1	0	0	1	0
C_{tm}	1	1	0	0	1	0
β	1	1	0	0	1	0
V_1	0	1	0	0	1	0
V_2	1	0	0	0	1	0
A_1	0	1	0	0	1	0
A_2	1	0	0	0	1	0
K_s	0	0	1	0	1	0
K_t	0	0	0	1	1	0
C_s	0	0	1	0	1	0
M_s	0	0	1	1	1	1
M_u	0	0	0	1	1	0

Table 3.3: MD-FSM at $a_1 = 0$, $a_3 = 1$

	r_1	r_2	r_3	r_4	D_b	I_b
C_d	1	1	0	0	1	0
w	1	1	0	0	1	0
C_{tm}	1	1	0	0	1	0
β	1	1	0	0	1	0
V_1	0	1	0	0	1	0
V_2	1	0	0	0	1	0
A_1	0	1	0	0	1	0
A_2	1	0	0	0	1	0
K_s	0	0	1	0	1	0
K_t	0	0	0	1	1	0
C_s	0	0	1	0	1	0
M_s	0	0	1	1	1	1
M_u	0	0	0	1	1	0

Table 3.4: MD-FSM at $a_1 = 1$, $a_3 = 0$

	r_1	r_2	r_3	r_4	D_b	I_b
C_d	1	1	0	0	1	0
w	1	1	0	0	1	0
C_{tm}	1	1	0	0	1	0
β	1	1	0	0	1	0
V_1	0	1	0	0	1	0
V_2	1	0	0	0	1	0
A_1	0	1	0	0	1	0
A_2	1	0	0	0	1	0
K_s	0	0	1	0	1	0
K_t	0	0	0	1	1	0
C_s	0	0	1	0	1	0
M_s	0	0	1	1	1	1
M_u	0	0	0	1	1	0

Table 3.5: MD-FSM at $a_1 = 0$, $a_3 = 0$

	r_1	r_2	r_3	r_4	D_b	I_b
C_d	1	1	0	0	1	0
w	1	1	0	0	1	0
C_{tm}	1	1	0	0	1	0
β	1	1	0	0	1	0
V_1	0	1	0	0	1	0
V_2	1	0	0	0	1	0
A_1	0	1	0	0	1	0
A_2	1	0	0	0	1	0
K_s	0	0	1	0	1	0
K_t	0	0	0	1	1	0
C_s	0	0	1	0	1	0
M_s	0	0	1	1	1	1
M_u	0	0	0	1	1	0

MATLAB Simulink is used to build the HBG model of the electro-hydraulic suspension system. In MATLAB Simulink, BG basic elements are built using the “subsystem” block in the Simulink library. For example, a R -element is established by set icon drawing command of the block as “disp(['R'])”. The constitutive relation of the R -element is built under the mask of the block. For the example of R : C_s in Figure 3.10, its constitutive relation described in (3.53) is modeled as shown in Figure 3.11. Similarly, as for other elements, such as transformer, gyrator and junctions, they are modeled in Simulink using the “subsystem” blocks.

Figure 3.11: Simulink model representing constitutive relation of R -element

The half arrow bond is also modeled using the “subsystem” block, the half arrow line and the stroke are drawn through the icon drawing command. For controlled junctions, they are modeled by common junctions with Boolean variables. For the example of controlled junction 0_{a1} in Figure 3.10, the constitutive relation in Simulink is depicted in Figure 3.12. Finally, all components are connected together by bonds to form a complete HBG as shown in Figure 3.13.

Figure 3.12: Simulink model of constitutive relation of 0-junction

Figure 3.13: HBG model of Figure 3.9 in Simulink environment

The nominal parameters of the electro-hydraulic suspension are listed in Table 3.6. The input is $u = 2 \sin(\pi t_s)$, sampling time t_s is 0.005 second. A fault occurrence is considered in unsprung spring K_t , with parameter value changes from 190000N/m to 95000N/m, which can also be taken as the fault of flat tire. Two fault scenarios have been simulated. In the first case, the road profile undergoes from random road to bumpy road, featuring a_1 changing from 1 to 0, the transition time is set at 7 second during the observation window. In the second case, the road profile undergoes from bumpy road to random road, featuring a_1 changing from 0 to 1, the transition time is set at 8 second during the observation window. Figure 3.14 and Figure 3.15 show the road profiles according to these two scenarios. The fault occurrence in unsprung spring K_t is introduced at 5s for both scenarios.

Table 3.6: Nominal physical parameters of the electro-hydraulic suspension

Parameter	Value	Unit
M_s	290	kg
M_u	59	kg
K_s	16800	N/m
K_t	190000	N/m
C_s	1000	N·s/m
A_1	3.8×10^{-4}	m ²
A_2	3.5×10^{-4}	m ²
V_1	1.2×10^{-4}	m ³
V_2	1.1×10^{-4}	m ³
P_s	1.5×10^7	Pa
C_{im}	0.3×10^{-12}	m ³ ·s ⁻¹ /Pa
ρ	900	kg·m ⁻³
β	1.2×10^9	Pa
C_d	0.6	/
w	1.5×10^{-3}	m
V	12.5	m/s
q_0	0.1	m ⁻¹
G_0	512×10^{-6}	m ³
A	0.05	m
L	5	m

Figure 3.14: Road profile of the first scenario

Figure 3.15: Road profile of the second scenario

Figure 3.16: Residual responses of the first scenario

Figure 3.16 and Figure 3.17 present the residual responses for these two scenarios. Note that the residual theoretically should be zero under normal condition. In practice, however, the residual is within certain error bounds as long as no faults occur during system operation. The value is not exactly zero over some time interval due to numerical inaccuracies caused by derivative approximation in Simulink and noise signal in road profile. Usually, the threshold is set by observation of system responses under normal condition. Misdetection means missing to detect the presence of an actually occurred fault and false alarm refers to an indication of fault which in fact does not happen. They are two contradictory factors when selecting the proper threshold for FDI purpose. Consideration of tradeoff is then made depending on experience and trial and error experiment. For the abrupt fault in this example, its effect on residual is larger than other factors and thus a simple fixed threshold is sufficient to detect the fault. The thresholds for these two scenarios are shown in Table 3.7.

Table 3.7: Thresholds for the two scenarios

	Th_1	Th_2	Th_3	Th_4
First scenario	0.5e-5	0.5e-5	0.9	0.15
Second scenario	0.3e-6	0.3e-6	0.5	0.12

These thresholds are set according to observe the residual responses during normal condition. Under normal operation, all residuals should below the thresholds. If at least one residual exceeds the threshold, the system is considered as faulty. It is observed that after 5s, the detected coherence vector is $[0 \ 0 \ 0 \ 1]$. From the MD-FSM for operating modes $\forall a_1 \in \{0,1\}$, both K_t and M_u share the same fault signatures which indicates the fault is non-isolable. Therefore, the targeted fault parameters are K_t and M_u . If p is chosen as 3, then T_1^1 , T_1^2 and T_1^3 are the switching time stamps for a_1 .

Figure 3.17: Residual responses of the second scenario

With an observation window of 5 seconds, one thousand of sample data ($N = 1000$) are collected after a fault is detected. Before proceeding to simulation comparison, careful selection of parameter settings is crucial to produce a competent result. The parameter sensitivity analysis is carried out where the inertia weight w_r of RPSO decreases from 0.8 to 0.5 with decrement of 0.1, and the $V_{\max,r}$ of BPSO increases from 4 to 6 with increment 1. In order to eliminate the influence of randomness, each parameter setting of AHPSO is carried out 50 times from different initial seeds of the random number to ensure the repetitiveness of convergence. It has been observed that the final solutions obtained with different parameter settings do not differ much (standard deviation are less than 1% of the mean value). Therefore, results presented are the mean of the 50 times. All the other parameters associated with AHPSO are taken as:

Particle number=50, Maximum iterations=150, $\Psi_1 = 2$, $\Psi_2 = 2$, $\eta = 0.5$.

During the optimization process, the parameters identified fall into the bounded ranges, including $K_t \in [50000 \ 300000] \text{N/m}$, $M_u \in [10 \ 150] \text{kg}$, $T_1^1 \in [0,5] \text{s}$, $T_1^2 \in [0,5] \text{s}$ and $T_1^3 \in [0,5] \text{s}$ [108]. The optimum solution is denoted as F_{fitness} in (3.15), so the bigger F_{fitness} is, the better the global search ability of the method is. Table 3.8 lists the results of parameter sensitivity analysis, and the best F_{fitness} is shown in boldface for all cases. According to these results, $w_r = 0.5$ for RPSO and $V_{\max,r} = 5$ for BPSO are chosen for the AHPSO algorithm of the first scenario and $w_r = 0.7$ for RPSO and $V_{\max,r} = 6$ for BPSO are chosen for the AHPSO algorithm of the second scenario. The different optimal parameter settings of AHPSO for different scenarios reveal that the parameter settings of AHPSO are actually problem dependent.

For comparison, AHPSO, GA and AGA [107] approaches have been tested on the same data collected after the fault occurring.

The parameters of GA are taken as:

Maximum iterations=150, Population = 50, $p_c = 0.6$, $p_m = 0.05$.

Table 3.8: Parameter sensitivity analysis for identification

		$V_{\max,r} = 4$	$V_{\max,r} = 5$	$V_{\max,r} = 6$
First scenario	$w_r = 0.8$	7.32	6.27	6.85
	$w_r = 0.7$	6.98	6.34	7.31
	$w_r = 0.6$	7.49	7.25	6.85
	$w_r = 0.5$	7.22	7.81	7.43
Second scenario	$w_r = 0.8$	8.71	8.63	8.70
	$w_r = 0.7$	8.25	7.68	8.92
	$w_r = 0.6$	8.51	8.46	7.98
	$w_r = 0.5$	7.87	8.14	8.42

The parameters of AGA are the same as those of GA, the adaptive scheme of AGA involves some rules to adjust adaptively the crossover and mutation rates according to the performance of the current genetic operators. It increases the probability of the genetic operator if it consistently produces a better offspring during the search process; on contrast, it reduces the probability of the genetic operator if it always produces a poorer offspring. This method can adaptively regulate the balance between the exploration and exploitation of the solution space [107].

Comparison results of all methods are shown in Table 3.9, it is seen that AHPSO is superior to the other two methods because AHPSO has better information sharing and conveying mechanism than GA and AGA. AHPSO also has good dynamics of balance between global and local search abilities. The results show the consistency between F_{fitness} and accuracy of identified parameters which means the highest F_{fitness} can provide the most accurate estimated parameters, except for M_u value in the second scenario, whereby the difference is so small that it can be neglected. Average execution CPU time for AHPSO approach is minimum, 27s for the first scenario and 25s for the second scenario, because it can easily adjust inertia weight of RPSO and maximum velocity of BPSO using simple “IF-THEN” rules. These results illustrate that the proposed method can identify the fault of K_t , the flat tire fault, and rule out M_u , the unsprung mass, as a fault; and that the proposed method can estimate the parameters accurately. The estimated switching time stamps also show a good matching with the implemented values.

Table 3.9: Comparison results using different methods

	methods	M_u	K_t	T_1^1	T_1^2	T_1^3	λ_1	F_{fitness}	CPU time (s)
First scenario	AHPSO	59.0052	95004	1.9990	5.00	5.00	1	7.81	27
	GA	58.7443	94946	1.9424	5.00	5.00	1	6.18	39
	AGA	58.9261	94982	1.9758	5.00	5.00	1	7.28	41
Second scenario	AHPSO	59.1024	94956	3.0002	5.00	5.00	0	8.92	25
	GA	58.6773	94839	2.9832	5.00	5.00	0	8.17	38
	AGA	59.0782	94877	2.9990	5.00	5.00	0	8.48	43

3.5 Case Study of an Electric Circuit

Figure 3.18: A hybrid system: an electric circuit

Figure 3.19: HBG of the electric circuit Figure 3.20: DHBG of the electric circuit

Figure 3.18 shows an example of hybrid system in electrical domain whose HBG is depicted in Figure 3.19. The system consists of three R elements $\{R_1, R_2, R_3\}$, three C elements $\{C_1, C_2, C_3\}$, one switch Sw_1 , one current (flow) sensor Df , and two voltage sensors $\{De_1, De_2\}$. The HBG has a 1-type controlled junction denoted by 1_c that represents the physical switch in electrical domain. The DHBG of the hybrid system is shown in Figure 3.20.

The constitutive relations of junctions 1_1 , 0_2 , 1_c , 0_4 , 1_5 and 0_6 are given by the following equations

$$e_1 - e_2 - e_3 - e_4 = 0 \quad (3.61)$$

$$f_4 - f_5 - af_6 = 0 \quad (3.62)$$

$$ae_6 - ae_7 - ae_8 = 0 \quad (3.63)$$

$$af_8 - f_9 - f_{10} - f_{11} = 0 \quad (3.64)$$

$$e_{11} - e_{12} - e_{13} = 0 \quad (3.65)$$

$$f_{13} - f_{14} - f_{15} = 0 \quad (3.66)$$

where variable a denotes the state of switch Sw_1 .

Three structurally independent GARRs can be generated from (3.62), (3.64) and (3.66) after eliminating the unknown variables. The unknown variables elimination process is achieved by following the causal path, from known to unknown, on the DHBG model.

In (3.62) the unknown variables f_4 , f_5 and f_6 can be calculated as follows

$$f_4 = Df \quad (3.67)$$

$$f_5 = C_1 \frac{d}{dt} e_5 = C_1 \frac{d}{dt} (V_{in} - R_1 Df) \quad (3.68)$$

$$f_6 = f_7 = \frac{1}{R_2} (V_{in} - R_1 Df - De_1) \quad (3.69)$$

The first GARR is obtained as follows

$$GARR_1 = Df - C_1 \frac{d}{dt} (V_{in} - R_1 Df) - \frac{a}{R_2} (V_{in} - R_1 Df - De_1) \quad (3.70)$$

The unknown variables f_8 , f_9 , f_{10} and f_{11} in (3.64) can be calculated as

$$f_8 = f_7 = \frac{1}{R_2}(V_{in} - R_1 Df - De_1) \quad (3.71)$$

$$f_9 = C_2 \frac{d}{dt} De_1 \quad (3.72)$$

$$f_{10} = 0 \quad (3.73)$$

$$f_{11} = f_{12} = \frac{1}{R_3}(De_1 - De_2) \quad (3.74)$$

The second GARR is obtained as follows

$$GARR_2 = \frac{a}{R_2}(V_{in} - R_1 Df - De_1) - C_2 \frac{d}{dt} De_1 - \frac{1}{R_3}(De_1 - De_2) \quad (3.75)$$

For (3.66) the unknown variables f_{13} , f_{14} and f_{15} can be calculated as

$$f_{13} = f_{12} = \frac{1}{R_3}(De_1 - De_2) \quad (3.76)$$

$$f_{14} = C_3 \frac{d}{dt} De_2 \quad (3.77)$$

$$f_{15} = 0 \quad (3.78)$$

The third GARR can be represented as follows

$$GARR_3 = \frac{1}{R_3}(De_1 - De_2) - C_3 \frac{d}{dt} De_2 \quad (3.79)$$

With these three GARRs, the MD-FSM tables for the circuit are shown as following

Table 3.10: MD-FSM at $a = 1$

	r_1	r_2	r_3	D_b	I_b
R_1	1	1	0	1	0
R_2	1	1	0	1	0
R_3	0	1	1	1	1
C_1	1	0	0	1	1
C_2	0	1	0	1	1
C_3	0	0	1	1	1

Table 3.11: MD-FSM at $a = 0$

	r_1	r_2	r_3	D_b	I_b
R_1	1	0	0	1	0
R_2	0	0	0	0	0
R_3	0	1	1	1	1
C_1	1	0	0	1	0
C_2	0	1	0	1	1
C_3	0	0	1	1	1

To implement the hybrid electrical system for verification of proposed fault identification algorithm, an experimental system that consists of the electrical hybrid system, MATLAB Simulink, and MATLAB Real-Time Window Target is built. Figure 3.21 depicts the experimental system where the high speed photomos

switching relay (AQV251) in Figure 3.22 is used to act as a switch. The nominal parameters of the hybrid system are:

$$R_1 = 135\Omega, R_2 = 135\Omega, R_3 = 135\Omega, C_1 = 10000\mu\text{F}, C_2 = 10000\mu\text{F}, C_3 = 10000\mu\text{F}.$$

Figure 3.21: Experiment setup

Figure 3.22: High speed photomos switching relay

The input $V_{in} = 5\sin(0.1t_s)$, sampling time t_s is 0.01s. A fault occurred in resistor R_1 from 135Ω to 67.5Ω at 5s is considered in the system, and the switching signal is depicted in Figure 3.23. The thresholds are set as: $Th_1=3e-4$, $Th_2=1e-3$, $Th_1=1e-4$.

Figure 3.23: Switching signal of the electric circuit

Figure 3.24: Residual responses of the electric circuit

Figure 3.24 shows the residual responses of the system. It is observed that after 5s, the detected coherence vector is $[1 \ 1 \ 0]$. From the Table 3.10, both R_1 and R_2 share the same fault signatures which indicates the fault is non-isolable. Therefore, the targeted fault parameters are R_1 and R_2 . If p is set as 3, then T_1^1 , T_1^2 and T_1^3 are the switching time stamps for a . The measurement in a 10s window is collected, and number of data $N = 1000$ is gathered after a fault is detected.

Similarly, the parameter sensitivity analysis is also carried out and results are recorded in Table 3.12, the best F_{fitness} is shown in boldface. According to the results, $w_r = 0.8$ for RPSO and $V_{\text{max},r} = 5$ for BPSO are chosen for the AHPSO algorithm.

Table 3.12: Parameter sensitivity analysis for identification

	$V_{\max,r} = 4$	$V_{\max,r} = 5$	$V_{\max,r} = 6$
$w_r = 0.8$	4.76	5.77	4.74
$w_r = 0.7$	3.21	5.60	5.05
$w_r = 0.6$	3.98	4.92	5.32
$w_r = 0.5$	3.64	4.86	5.37

Table 3.13 is the summary of the identification results for different methods, and the results are the average of 50 runs. Both AHPSO and AGA have similar results, but AHPSO uses less CPU time due to the simplicity of structure. The experimental results confirm that the proposed AHPSO is superior to GA and AGA in terms of identification accuracy and computational cost. $\lambda_1 = 1$ in Table 3.13 means that the initial mode of the controlled junction during the collected data is 1(ON state) which matches the designed value. These results illustrate that the proposed method can identify the fault of R_1 and rule out R_2 as fault.

Table 3.13: Comparison results using different methods

methods	R_1	R_2	T_1^1	T_1^2	T_1^3	λ_1	F_{fitness}	CPU time (s)
AHPSO	66.721	133.887	5.001	10.00	10.00	1	5.77	21
GA	65.315	132.257	5.176	10.00	10.00	1	4.23	33
AGA	66.572	133.163	5.087	10.00	10.00	1	5.15	36

3.6 Conclusions

This chapter presents an algorithm for simultaneous identification of fault parameters and mode switching stamps for hybrid systems. The deployment of GARRs makes it possible to formulate the system behavior in a unified form at all modes. For simultaneous identification of parameter fault and mode switching stamps, a unified formula is developed to cater for the difficult situation where mode switching occurs after the fault. An AHPSO algorithm is developed to search for the fault parameters, switching time sequence and IMC using RPSO and BPSO simultaneously. The critical parameters of the algorithm are dynamically tuned by the individual level adaptive scheme to improve the system performance. Case study results of different energy domains show the effectiveness and efficiency of the proposed algorithm.

Chapter 4

Multiple Fault Diagnosis without Fault Pattern and Mode Change Information

Multiple fault diagnosis received a lot of attention during recent years [75]. Since a complex system includes many components and subsystems, several faults may happen simultaneously or one after another. Fault isolation technique under single fault assumption could be misleading when it is applied to monitor the system with multiple faults. The nature of faults in a monitored system can be classified as abrupt fault, incipient fault and intermittent fault. Fault identification under abrupt fault assumption, i.e. constant values for fault parameters, is not appropriate to handle cases related to incipient fault and intermittent fault. It is difficult to carry out the fault estimation task if the fault nature information is lack. The situation is further complicated when the mode information is unavailable.

This chapter deals with the problem of multiple fault diagnosis with unknown fault pattern and mode change. The faults could be happen either in a detectable mode or in a non-detectable mode in a monitored hybrid system. A multiple fault isolation algorithm is introduced when the system is lack of fault pattern and mode change information. A hypothesis set which includes both suspected faults and suspected mode changes are established after fault isolation. Fault estimation is realized by differential evolution algorithm. Simulation and experiment results are reported to validate the proposed method.

4.1 Types of Fault

Faults are defined as deviation from the normal behavior in the system or its instrumentation and are categorized as follows:

1) Component faults

Component faults are resulted from changes in the parameters of system components. Examples are internal leakage of the cylinder, loosen belt and blockage of pipe.

2) Sensor faults

Sensor faults happen when there are discrepancies between measured signals and their actual values.

3) Actuator faults

Actuator faults occur when there are discrepancies between input command of an actuator and its actual output to the system.

In general, the nature of faults can be classified as three categories:

1) Abrupt faults

Abrupt faults are step like and persistent, which are usually modeled as instantaneous changes in system parameters and lead to distinct inconsistency in the monitored system.

2) Incipient faults

Incipient faults are slowly developing and usually describe the wear and ageing of system components. They are characterized by gradual drifts in the corresponding component parameters from the time instance of fault occurrence. Incipient faults are relatively difficult to handle due to the slowly developing nature of the faults and the compensation of the feedback control.

3) Intermittent faults

Intermittent faults are last for only very short period of time, but may sometimes cause disastrous results. They are hard to handle due to their difficult anticipation.

Uncertainties and disturbances are two main imperfections of a system and they are usually out of the scope of fault diagnosis.

1) Uncertainties

Uncertainties in modeling can be due to a bad estimation of dynamics in a system,

a non-precise identification of the numerical values of the parameters or variation of their values because of heat, time or working conditions.

2) Disturbances

Disturbances usually refer to noises in sensor measurements that are high frequency signals. Other factors like unmeasured friction and backlash are also considered as disturbances.

A good diagnostic system should be robust to various disturbances and uncertainties.

4.2 Augmented GARRs for Fault Diagnosis

In this work, the faults are divided into two categories: parametric fault and nonparametric fault. Parametric fault is characterized by the changing of component parameter. For example, blockage of pipe is a typical example of incipient parametric fault and the blockage, often modeled as resistance, usually increases over time. Nonparametric fault refers to the deterioration of system components that are under monitoring but cannot be described by physical parameters. Examples include sensors and actuators. In order to quantify the severity of the fault, an efficiency factor β is introduced, where $\beta = 0$ indicates a total failure while $\beta = 1$ denotes no fault, and $0 < \beta < 1$ represents a fault. In other words, $1 - \beta$ indicates the severity of the fault. When the system is under normal condition, all β are equal to 1, when a fault occurs, β related to the suspected fault candidates are not equal to 1 anymore and can be treated as unknown parameters in the dynamic model. In the HBG, actuator fault is modeled by multiplying the input by β , and sensor fault is modeled by multiplying the normal measurement by β , then in normal condition, all β are known and equal to 1, when fault happens, those β related to the suspected fault candidates become unknown parameters. Based on this spirit, a set of Augmented GARRs (AGARRs) is proposed to extend the capability of GARRs to identify nonparametric fault, and AGARR equations take the following form

$$\bar{g}_l(\theta, \alpha, \beta, De, Df, u) = 0 \text{ for } l = 1, 2, \dots, m \quad (4.1)$$

where $\beta = [\beta_1, \beta_2, \dots, \beta_g]^T$, g is the total number of nonparametric components.

It is worth to note that in the AGARR equations described in (4.1), De and Df represent sensors measurement, they can be faulty signals or faulty free signals, since sensor faults are introduced by multiplying the normal measurement by β , then in the AGARR equations, normal measurement of sensors are represented by dividing De and Df by β . In normal condition, all β are known and equal to 1, when the total sensor failure occurs and De or Df is equal to 0 during observation window, all β related to the suspected fault candidates (De or Df) are equal to 0. For the other cases, i.e. partial failure with $0 < \beta < 1$, those β related to the suspected fault candidates (De or Df) become unknown parameters.

4.3 Multiple Fault Isolation without Fault Pattern and Mode Change Information

For multiple fault diagnosis of hybrid systems with unknown mode changes and fault pattern, the hypotheses set generation during fault isolation is much complex, since the fault may develop at a non-detectable mode and be detected at a detectable mode, the combinational effect of mode change and fault can result in the inconsistency in the health monitoring process. For example, let us consider an example of two Mode Dependent-FSM (MD-FSM), analogous to FSM for continuous system FDI [82], shown in Table 4.1 and Table 4.2, in which $\alpha \in \{0,1\}$ represents the operating mode of the hybrid system; θ_i , $i=1,\dots,4$, denote the system parameters; β_i , $i=1,\dots,4$, represent the efficiency factors related to sensors or actuators; r_i , $i=1,\dots,3$, denote the three residuals. The MCSM of the system is shown in Table 4.3.

The initial mode is $a = 0$ and is predicted by mode tracker. The mode change is set at 5 second which is unknown for the designer. When residual r_3 is abnormal at 5 second, it leads to a coherence vector $[0 \ 0 \ 1]$. The mode tracker cannot detect a new mode, so the inconsistency is not caused by a single mode change of a , denoted by $\bar{a}$. From Table 4.1, the inconsistency could be due to a change of θ_2 . From Table 4.2 and Table 4.3, $\bar{a}$ could make θ_3 and β_1 to be detectable if θ_3 and β_1 develop at $a = 0$, and

Table 4.1: MD-FSM at $a = 0$

	r_1	r_2	r_3	D_b
θ_1	1	0	0	1
β_1	0	0	0	0
θ_2	0	0	1	1
β_2	0	1	0	1
θ_3	0	0	0	0
β_3	0	1	0	1
θ_4	0	1	1	1
β_4	0	1	0	1

Table 4.2: MD-FSM at $a = 1$

	r_1	r_2	r_3	D_b
θ_1	1	0	0	1
β_1	0	0	1	1
θ_2	0	0	1	1
β_2	0	1	0	1
θ_3	0	0	1	1
β_3	0	1	0	1
θ_4	0	1	1	1
β_4	0	1	0	1

Table 4.3: MCSM of the example

C-J	r_1	r_2	r_3	D_b	I_b
a	0	0	1	1	1

the combinational effect of mode change and fault may result in the inconsistency. Therefore, the hypothesis set could be $\delta = \{\theta_2, \beta_1 \& \bar{a}, \theta_3 \& \bar{a}, \beta_1 \& \theta_3 \& \bar{a}\}$. Since the fault pattern is unknown in advance, the fault pattern hypothesis set after fault isolation could be

$$\delta = \{\theta_2', \theta_2^A \beta_1' \& \bar{a}, \beta_1^A \& \bar{a}, \theta_3' \& \bar{a}, \theta_3^A \& \bar{a}, \beta_1' \& \theta_3' \& \bar{a}, \beta_1^A \& \theta_3^A \& \bar{a}, \beta_1' \& \theta_3^A \& \bar{a}, \beta_1^A \& \theta_3' \& \bar{a}\}$$

where θ^A / β^A denotes abrupt fault and θ' / β' represents incipient fault [109]. Note that the number of element in the fault pattern hypothesis set will increase notably when the length of element in hypothesis set increase, e.g. the length of the element $\theta_1 \& \theta_2 \& \dots \& \theta_n$ in hypothesis set is n , the corresponding number of element in the fault pattern hypothesis set is 2^n . As a result, the computational complexity of the algorithm will grow significantly with the increase of n . This drawback can be overcome by defining a binary variable η for each fault candidate in the hypothesis set. When η takes value of 1, it represents incipient fault; and when η is equal to 0, it denotes abrupt fault. In this way, the fault pattern estimation is transformed to binary variable selection in the identification process.

Since mode change may occur after a fault is detected and model dependent mode tracker is not useful, it is difficult to carry out fault identification if the mode information is unavailable. The mode change information is used to determine which

model the system is in. In other words, under different modes the system is governed by different AGARRs, as shown in Mode-AGARR Table (MAT). When the mode is determined, the corresponding AGARRs can be utilized for fault identification. A unified formula is adopted to describe the mode change as described in (3.4) in Chapter 3.

4.4 Fault Estimation via Differential Evolution

The formula in (3.4) is indeed a discontinuous function, which might be problematic if analytical methods, such as least square method and Gauss-Newton method, are used for the switching time identification considering the non-differentiable function. Furthermore, binary number η representing different fault pattern is not involved in the objective function to be minimized, thus no gradient information can be obtained regarding η . Therefore, heuristics-based Evolutionary Algorithms (EAs) can be efficient alternatives.

In this study, a Hybrid DE (HDE) algorithm is developed, in which Real DE (RDE) is utilized to identify the switching time stamps, fault parameter value for abrupt fault and dynamical model coefficient for incipient fault, and Binary DE (BDE) is adopted to estimate η representing fault pattern. Assume that the incipient fault follows the dynamic behavior of $\dot{\theta} = a \cdot \theta$, so if θ_i is the true incipient fault, then the AGARR $\bar{g}_i(\theta, n)$ that contains θ_i can be reformulated as

$$\bar{g}_i(sol, \theta_j, n), \text{ with } sol = \theta_i(0) \exp(a \cdot t) \quad (4.2)$$

$$t = n \cdot t_s, \quad j = 1, \dots, m \text{ and } j \neq i$$

where t_s is the sampling period. Assume the mode is persistent between the end of FDI and the beginning of data collection. Note that in the solution the initial condition $\theta_i(0)$ is not equal to the nominal value of θ_i if the fault in the hypothesis set initiates at the non-detectable mode, therefore, $\theta_i(0)$ is also an unknown coefficient which needs to be estimated.

The RDE and BDE evolve simultaneously and are coupled through the common fitness function defined as follow

Figure 4.1: Individual structure of HDE algorithm considering multiple faults

Figure 4.2: A model based health monitoring framework for hybrid systems

$$F_{fitness} = 1 / (\sum_{l=1}^m \sum_{n=1}^N |\bar{g}_l^n| + \varepsilon) \quad (4.3)$$

where ε is a small positive constant which is used to avoid zero division during the search process.

The individual structure for one HDE is shown in Figure 4.1, where n is the number of faults in the element in the hypothesis set, e.g. for the element β_1 & θ_3 & $\bar{a}$, n is equal to 2. The complete health monitoring framework, which includes HBG modeling scheme, the MCSM based mode tracker, the AGARRs based fault detection

and isolation mechanism, and the Multiple Hybrid Differential Evolution (MHDE) based multiple fault identification module is shown in Figure 4.2.

4.5 Illustrative Example

4.5.1 System Description

Figure 4.3: An electrical hybrid system

Figure 4.3 illustrates an example of hybrid system in electrical domain whose DHBG is obtained using SCAPH and is depicted in Figure 4.4. The system consists of two switches $\{sw_1, sw_2\}$ and nine components $\{R_1, R_2, R_3, R_4, R_5, C_1, C_2, V_1, V_2\}$. De_1 and De_2 stand for the voltage (effort) sensors which measure the respective voltages of the circuit. Df represents the current (flow) sensor. In Figure 4.4, 1_1 and 1_6 are controlled junctions which represent two switches. Two Boolean variables a_1 and a_2 are used to describe the state (ON/OFF) of the controlled junctions.

Figure 4.4: DHBG of the circuit

First, consider the constitutive relation of junction 0_2 ,

$$a_1 f_3 + f_4 - f_7 - f_8 = 0 \quad (4.4)$$

Then tracking back the causal paths yields

$$\begin{aligned}
 f_3 = f_2 &= \frac{1}{R_1} e_2 = \frac{1}{R_1} (V_1 - De_1) \\
 f_4 = f_5 &= \frac{1}{R_2} e_5 = \frac{1}{R_2} (e_6 - e_4) = \frac{1}{R_2} (V_2 - De_1) \\
 f_7 &= C_1 \frac{d}{dt} e_7 = C_1 \frac{d}{dt} De_1 \\
 f_8 &= Df
 \end{aligned} \tag{4.5}$$

Substituting (4.5) into (4.4) gives

$$AGARR_1 = \frac{a_1}{R_1} (\beta_{V_1} V_1 - \frac{De_1}{\beta_{De_1}}) + \frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} (\frac{De_1}{\beta_{De_1}}) - \frac{Df}{\beta_{Df}} = 0 \tag{4.6}$$

Next, consider junction 1₄, the constitutive relation of this junction can be expressed as

$$e_8 - e_9 - e_{10} = 0 \tag{4.7}$$

By following the causal paths, the effort variables e_8 , e_9 and e_{10} can be represented as

$$\begin{aligned}
 e_8 &= De_1 \\
 e_9 &= R_3 \cdot Df \\
 e_{10} &= De_2
 \end{aligned} \tag{4.8}$$

The second AGARR can be obtained by combining (4.8) and (4.7)

$$AGARR_2 = \frac{De_1}{\beta_{De_1}} - R_3 \frac{Df}{\beta_{Df}} - \frac{De_2}{\beta_{De_2}} = 0 \tag{4.9}$$

Finally consider junction 0₅, with the constitutive relation

$$f_{10} - f_{11} - f_{12} - a_2 f_{13} = 0 \tag{4.10}$$

Tracking the casual paths, gives

$$\begin{aligned}
 f_{10} &= Df \\
 f_{11} &= C_2 \frac{d}{dt} e_{12} = C_2 \frac{d}{dt} De_2 \\
 f_{12} &= \frac{1}{R_4} e_{12} = \frac{1}{R_4} De_2 \\
 f_{13} = f_{14} &= \frac{1}{R_5} e_{14} = \frac{1}{R_5} De_2
 \end{aligned} \tag{4.11}$$

Then AGARR₃ is given by

$$AGARR_3 = \frac{Df}{\beta_{Df}} - C_2 \frac{d}{dt} \left(\frac{De_2}{\beta_{De_2}} \right) - \frac{1}{R_4} \frac{De_2}{\beta_{De_2}} - \frac{a_2}{R_5} \frac{De_2}{\beta_{De_2}} = 0 \quad (4.12)$$

From these three AGARRs, the MCSM can be established in Table 4.4. A MAT, as shown in Table 4.5, is generated from the AGARRs, to present the ARR at each mode with $[a_1 \ a_2]$. The MD-FSM tables can be obtained from the MAT and are shown in Table 4.6 ~ Table 4.9.

Table 4.4: The MCSM of the electrical circuit

C-J	r_1	r_2	r_3	D _b	I _b
a_1	1	0	0	1	1
a_2	0	0	1	1	1

Table 4.5: The MAT of the electrical circuit

Mode	r_1	r_2
[0 0]	$\frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} \left(\frac{De_1}{\beta_{De_1}} \right) - \frac{Df}{\beta_{Df}} = 0$	$\frac{De_1}{\beta_{De_1}} - R_3 \frac{Df}{\beta_{Df}} - \frac{De_2}{\beta_{De_2}} = 0$
[0 1]	$\frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} \left(\frac{De_1}{\beta_{De_1}} \right) - \frac{Df}{\beta_{Df}} = 0$	$\frac{De_1}{\beta_{De_1}} - R_3 \frac{Df}{\beta_{Df}} - \frac{De_2}{\beta_{De_2}} = 0$
[1 0]	$\frac{1}{R_1} (\beta_{V_1} V_1 - \frac{De_1}{\beta_{De_1}}) + \frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} \left(\frac{De_1}{\beta_{De_1}} \right) - \frac{Df}{\beta_{Df}} = 0$	$\frac{De_1}{\beta_{De_1}} - R_3 \frac{Df}{\beta_{Df}} - \frac{De_2}{\beta_{De_2}} = 0$
[1 1]	$\frac{1}{R_1} (\beta_{V_1} V_1 - \frac{De_1}{\beta_{De_1}}) + \frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} \left(\frac{De_1}{\beta_{De_1}} \right) - \frac{Df}{\beta_{Df}} = 0$	$\frac{De_1}{\beta_{De_1}} - R_3 \frac{Df}{\beta_{Df}} - \frac{De_2}{\beta_{De_2}} = 0$
	r_3	
[0 0]	$\frac{Df}{\beta_{Df}} - C_2 \frac{d}{dt} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_4} \frac{De_2}{\beta_{De_2}} = 0$	
[0 1]	$\frac{Df}{\beta_{Df}} - C_2 \frac{d}{dt} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_4} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_5} \frac{De_2}{\beta_{De_2}} = 0$	
[1 0]	$\frac{Df}{\beta_{Df}} - C_2 \frac{d}{dt} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_4} \frac{De_2}{\beta_{De_2}} = 0$	
[1 1]	$\frac{Df}{\beta_{Df}} - C_2 \frac{d}{dt} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_4} \frac{De_2}{\beta_{De_2}} - \frac{1}{R_5} \frac{De_2}{\beta_{De_2}} = 0$	

Table 4.6: MD-FSM at $[a_1 \ a_2] = [1 \ 1]$ Table 4.7: MD-FSM at $[a_1 \ a_2] = [0 \ 1]$

$[1 \ 1]$	r_1	r_2	r_3	D_b
β_{V_1}	1	0	0	1
β_{V_2}	1	0	0	1
C_1	1	0	0	1
C_2	0	0	1	1
R_1	1	0	0	1
R_2	1	0	0	1
R_3	0	1	0	1
R_4	0	0	1	1
R_5	0	0	1	1
β_{Df}	1	1	1	1
β_{De_1}	1	1	0	1
β_{De_2}	0	1	1	1

$[1 \ 0]$	r_1	r_2	r_3	D_b
β_{V_1}	0	0	0	0
β_{V_2}	1	0	0	1
C_1	1	0	0	1
C_2	0	0	1	1
R_1	0	0	0	0
R_2	1	0	0	1
R_3	0	1	0	1
R_4	0	0	1	1
R_5	0	0	1	1
β_{Df}	1	1	1	1
β_{De_1}	1	1	0	1
β_{De_2}	0	1	1	1

Table 4.8: MD-FSM at $[a_1 \ a_2] = [1 \ 0]$ Table 4.9: MD-FSM at $[a_1 \ a_2] = [0 \ 0]$

$[0 \ 1]$	r_1	r_2	r_3	D_b
β_{V_1}	1	0	0	1
β_{V_2}	1	0	0	1
C_1	1	0	0	1
C_2	0	0	1	1
R_1	1	0	0	1
R_2	1	0	0	1
R_3	0	1	0	1
R_4	0	0	1	1
R_5	0	0	0	0
β_{Df}	1	1	1	1
β_{De_1}	1	1	0	1
β_{De_2}	0	1	1	1

$[0 \ 0]$	r_1	r_2	r_3	D_b
β_{V_1}	0	0	0	0
β_{V_2}	1	0	0	1
C_1	1	0	0	1
C_2	0	0	1	1
R_1	0	0	0	0
R_2	1	0	0	1
R_3	0	1	0	1
R_4	0	0	1	1
R_5	0	0	0	0
β_{Df}	1	1	1	1
β_{De_1}	1	1	0	1
β_{De_2}	0	1	1	1

4.5.2 Simulation Results

There are two fault scenarios, the first one is studied through simulation and the second one is verified through experiment. In simulation study, the physical parameters of the circuit are: $V_1 = 7\text{V}$, $V_2 = 3\text{V}$, $R_1 = 670\Omega$, $R_2 = 215.6\Omega$, $R_3 = 67.5\Omega$, $R_4 = 215.4\Omega$, $R_5 = 509\Omega$, $C_1 = 10000\mu\text{F}$, $C_2 = 4700\mu\text{F}$, sampling time t_s is 0.05s , $\varepsilon = 0.01$ in (4.3). Two abrupt faults are introduced. The first one is a resistor fault in R_4 whose faulty value is 50Ω ; the other one is a capacitor fault in C_2 with faulty value $10000\mu\text{F}$. The fault profiles are presented in Figure 4.5. The switches programming is shown in Figure 4.6. Both the switches programming and fault pattern are unknown for the designer.

Figure 4.5: Fault profiles for simulation

Figure 4.6: Switches programming for simulation

Figure 4.7 demonstrates the simulation responses of AGARRs for the first fault scenario. The first three graphs in the figure are the residual responses, in which the dash line denotes the threshold with value 0.001. These thresholds are set by observing the normal residual responses. The last graph depicts the actual mode versus the mode tracking result (dashed). The first step is the system initial mode

identification. At this stage, all of the ARR_s in Table 4.5 are running simultaneously and mode 3, i.e. $[a_1 \ a_2] = [1 \ 1]$, is identified. Notice that at the first few seconds, modes are much alike due to the initial condition $V_{C_1} = V_{C_2} = 0$ and mode identification takes longer time. At $t = 4\text{s}$, an inconsistency invokes the mode tracker. According to the coherence vector $[0 \ 0 \ 1]$ and the MCSM in Table 4.4, since the last known mode is 3, mode 1 (a change of a_2) is suspected and identified by the mode tracker. At $t = 8\text{s}$, the coherence vector is $[1 \ 0 \ 0]$, since the last known mode is 1, mode 0 (a change of a_1) is suspected and identified by the mode tracker. At $t = 13\text{s}$, the coherence vector is $[1 \ 0 \ 0]$, since the last known mode is 0, mode 1 (a change of a_1) is suspected and identified by the mode tracker. This terminates the mode tracking process of the normal stage. At $t = 15\text{s}$, two parametric faults are injected which leads to the coherence vector $[0 \ 0 \ 1]$, but a new mode is not identified by the mode tracker.

Figure 4.7: Simulation results for the first scenario

Since the last known mode is 1, according to the MD-FSM in Table 4.8, the inconsistency may be due to the fault in R_4 or C_2 or their combination. From Table 4.8, since R_5 is non-detectable at mode 1, a change of a_2 will make the fault in R_5 to be detectable, and the combinational effect of mode change and fault occurrence can cause the observed inconsistency. Consequently, a hypothesis set $\{C_2, R_4, C_2 \& R_4, \bar{a}_2 \& R_5\}$ is generated. All observations have been captured from the moment that a fault is detected. With an observation window of 5 seconds, $N=100$ sample data are collected. Next step is to choose the true faults with correct pattern among a set of fault candidates in the hypothesis set with unknown mode change.

The fault identification module is activated in which number of four HDE estimators are running in parallel, where p is chosen as 3 in (3.6) and (3.7). The parameters associated with HDE are chosen as: Population size=300, Maximum iterations=200, $F=0.4$, $C_r=0.8$. Mutation mechanism DE/rand/1/bin is used.

Figure 4.8: Mean fitness evolution versus iterations for simulation

In order to eliminate the effect of randomness, the HDE is carried out 20 times from different initial seeds of the random number to ensure the repetitiveness of convergence. It has been observed that the final solutions with different parameter settings do not differ much (standard deviation are less than 3% of the mean value). Consequently, results presented are the mean of the 20 times. Figure 4.8 presents the evolution of mean fitness value versus iterations. It is obvious that the true faults are C_2 & R_4 . Table 4.10 summarizes the identification results, the estimated fault pattern for C_2 & R_4 is $[\eta_{C_2} \quad \eta_{R_4}] = [0 \quad 0]$, which means they are both abrupt faults. During the estimation process, only GARR₃ which contains a_2 is used in the fitness function in (4.3) and the switching time stamps of a_2 are $T_2^1(s)$, $T_2^2(s)$ and $T_2^3(s)$. The switching time stamps are accurately identified from Table 4.10. The simulation results show that the proposed identification algorithm can simultaneously estimate the fault pattern, switching time stamps and fault parameter values for hybrid system.

Table 4.10: Summary for simulation results

	$C_2(\mu\text{F})$	$R_4(\Omega)$	η_{C_2}	η_{R_4}	$T_2^1(s)$	$T_2^2(s)$	$T_2^3(s)$
Designed values	10000	50	0	0	3	5	5
Estimated values	9998.2	49.8	0	0	2.992	5	5

4.5.3 Experiment Results

A circuit test-bed is established to verify the proposed method. The real time FDI is based on the MATLAB Real-Time Windows Target, and two data acquisition cards (NI PCI-6025E and NI PCI-6713). The switches are implemented by two PhotoMOS relays (AQV251) and controlled by the real time software. The physical parameters of the circuit are: $V_1 = 7\text{V}$, $V_2 = 9\text{V}$, $R_1 = 655.3\Omega$, $R_2 = 216\Omega$, $R_3 = 67.6\Omega$, $R_4 = 215.4\Omega$, $R_5 = 510\Omega$, $C_1 = 10000\mu\text{F}$, $C_2 = 4700\mu\text{F}$, sampling time t_s is 0.05 s. In normal condition, all efficiency factors are equal to 1, i.e. $\beta_{V_1} = 1$, $\beta_{V_2} = 1$, $\beta_{De_1} = 1$, $\beta_{De_2} = 1$ and $\beta_{Df} = 1$. An actuator fault occurs when there is discrepancy between command input of an actuator and its actual output to system. It is introduced by

reducing β_{V_1} at moment $t=9s$ as shown in Figure 4.9, in which β_{V_1} gradually decreases with dynamical model coefficient $a = -0.02$. This fault is realized by multiplying input V_1 by β_{V_1} in Simulink under Real-Time Windows Target environment. Another abrupt fault is introduced in R_1 with faulty value 51.2Ω . Both faults develop at a non-detectable mode. The fault profiles are presented in Figure 4.9. The switches programming is shown in Figure 4.10. In order to filter out the influence of measurement noise on the residuals, a low pass filter $\frac{15}{s+15}$ is adopted.

Figure 4.9: Fault profiles for experiment

Figure 4.10: Switches programming for experiment

Figure 4.11 presents the experiment responses for the second fault scenario. For initial mode identification, all of the ARR_s in Table 4.5 are running simultaneously and mode 3, i.e. $[a_1 \ a_2] = [1 \ 1]$, is identified. At $t = 4s$, an inconsistency invokes the mode tracker. According to the coherence vector $[0 \ 0 \ 1]$ and the MCSM in Table 4.4, since the last known mode is 3, mode 1 (a change of a_2) is suspected and identified by the mode tracker. At $t = 8s$, the coherence vector is $[1 \ 0 \ 0]$, since the last known mode is 1, mode 0 (a change of a_1) is identified. At $t = 12s$, the coherence vector is $[1 \ 0 \ 0]$, but a new mode is not identified by the mode tracker. Since the

last know mode is 0, according to the MD-FSM in Table 4.9, the fault in β_{V_2} or C_1 or R_2 or their combination can cause this inconsistency. From Table 4.9, since β_{V_1} and R_1 are non-detectable at mode 0, a change of a_1 will make the fault in β_{V_1} and R_1 to be detectable, and the combinational effect of mode change and fault may lead to the inconsistency. After fault isolation module, a hypothesis set including both suspected mode and faults is generated as

$$\{\beta_{V_2}, C_1, R_2, \beta_{V_2} \& C_1, \beta_{V_2} \& R_2, C_1 \& R_2, \beta_{V_2} \& C_1 \& R_2, \bar{a}_1 \& \beta_{V_1}, \bar{a}_1 \& R_1, \bar{a}_1 \& \beta_{V_1} \& R_1\}$$

any of the element happening supports the inconsistency.

Figure 4.11: Experiment results for the second scenario

All observations have been captured from the moment that a fault is detected. With an observation window of 8 seconds, $N = 160$ sample data are collected. The fault identification module is invoked in which number of ten HDE estimators are running in parallel, where p is chosen as 3 in (3.6) and (3.7). The control parameters in HDE are chosen as: Population size=300, Maximum iterations=200, $F = 0.35$, $C_r = 0.7$. Figure 4.12 depicts the evolution of mean fitness value versus iterations. It is known

that the true faults are in $\bar{a}_1$ & β_{V_1} & R_1 . Table 4.11 lists the identification results, in which $\beta_{V_1}(0)$ is the initial value of β_{V_1} in the measurement. Since the fault in β_{V_1} is introduced at $t = 9s$ and is detected at $t = 12s$, the initial value of β_{V_1} , i.e. $\beta_{V_1}(0)$, is not equal to 1 in the collected data. The fault pattern estimated for β_{V_1} is $\eta_{\beta_{V_1}} = 1$ which indicates it is an incipient fault, while the fault pattern of R_1 is $\eta_{R_1} = 0$ which means it is an abrupt fault. During the estimation process, only GARR₁ which contains a_1 is used in the fitness function in (4.3) and the switching time stamps of a_1 are $T_1^1(s)$, $T_1^2(s)$ and $T_1^3(s)$. The results confirm that the MHDE based multiple fault identification method works well in real experiment.

Figure 4.12: Mean fitness evolution versus iterations for experiment

Table 4.11: Summary for experiment results

	$\beta_{V_1}(0)$	a	$R_1(\Omega)$	$\eta_{\beta_{V_1}}$	η_{R_1}	$T_1^1(s)$	$T_1^2(s)$	$T_1^3(s)$
Designed values	0.9417	-0.02	51.2	1	0	6	8	8
Estimated values	0.9427	-0.021	51.9	1	0	5.96	8	8

4.6 Conclusions

In this chapter, a multiple fault identification method for hybrid system with unknown mode change and fault pattern is proposed. The fault can either develop at a detectable mode, or at a non-detectable mode. Since the fault pattern is unknown beforehand, a multiple fault isolation algorithm is developed, in which the fault pattern of each fault candidate is represented by a binary variable. A unified formula is utilized to describe the mode change. Once the hypothesis set is generated after fault isolation module, a MHDE algorithm is proposed for multiple fault identification. In the MHDE, each HDE estimator is adopted to estimate the fault pattern, mode change indexes and abrupt fault parameter value/incipient fault dynamic model coefficient. Simulation and experiment results of different fault scenarios validate the proposed algorithms.

Chapter 5

Prognosis of Hybrid Systems with Multiple Incipient Faults

Condition-Based Maintenance (CBM) works based on condition measurements and does not interrupt normal system operation. It helps avoid excessive or insufficient maintenance and ultimately results in higher system availability. Diagnosis and prognosis are two important aspects in CBM. Diagnosis is posterior event analysis and prognosis is prior event analysis. Prognosis is much more efficient than diagnostics to achieve zero-downtime performance. This is very important when a fault or a failure is catastrophic in some situations (e.g. nuclear power plant and helicopter gearbox).

This chapter aims to develop a model based prognosis method for hybrid systems with multiple incipient faults. Multiple incipient faults are considered in a complex hybrid system and these faults can develop during a mode when the faults are not detectable. The unknown degradation characteristics of each incipient fault is identified to fit into one of the four possible dynamic behaviors and such a degradation model will serve as a base for prognosis purpose.

5.1 Fundamentals of Prognosis

Recently, maintenance policies have evolved from early reactive maintenance, to age-based preventive maintenance, then to nowadays condition-based maintenance. Reactive maintenance is carried out usually after the system breakdown. In order to prevent emergency shutdowns and catastrophic failures, age-based preventive maintenance is introduced where maintenance is carried out based on system

operating time regardless of the health condition of a machine. Age-based preventive maintenance may sometimes reduce unexpected failures, but it is not cost effective and cannot eliminate catastrophic failures. Consequently, these conventional maintenance strategies do not fulfill the needs of high reliability and better quality industrial equipments. Fortunately, CBM can be an effective alternative where it attempts to avoid unnecessary maintenance by taking maintenance actions only when there is evidence of abnormality in the monitored system.

Three key elements of effective CBM are data acquisition (i.e. the collection of machine health information), data processing (i.e. the conditioning and feature extraction/selection of acquired data) and decision making (i.e. the recommendation of maintenance actions through diagnosis and/or prognosis). Today's concept of machine diagnosis comprises the automated detection and classification of faults, whereas machine prognosis is the automated estimation of how soon and likely a failure will occur. A CBM program can be used to do diagnostics or prognostics, or both. No matter what the objective of a CBM program is, however, the above three CBM steps are followed.

The word “prognosis” indicates the foretelling of the probable course of a disease, a term widely used in medical practice. In the industrial and manufacturing arenas, prognosis is the capability to provide early detection and isolation of precursor and/or incipient fault condition to a component failure condition and to have the technology and means to manage and predict the progression of this fault condition to component failure.

The time left before observing a failure is usually called RUL. RUL, also called remaining service life, residual life or remnant life, refers to the time left before observing a failure given the current machine age and condition, and the past operation profile. It is defined as the conditional random variable:

$$T - t | T > t, Z(t) \quad (5.1)$$

where

- T random variable of time to failure (TTF);
- t current age;

$Z(t)$ past condition profile up to the current time.

Many industrial systems exhibit increasing wear and tear of equipment during operation. For example, an automobile has many components, such as the suspension, gear box, and valves which exhibit different types of performance degradation due to erosion, friction, internal leakage, and cracks. Prognosis is viewed as an add-on capability to diagnosis. It is based on the analysis of failure modes, detection of early signs of wear and aging, and fault conditions. These signs are then correlated with a damage propagation model. The discipline that links studies of failure mechanisms to system lifecycle management is often referred to as Prognostics and Health Management (PHM). In recent years, PHM has emerged as one of the key enablers for achieving efficient system-level maintenance and lowering life-cycle costs.

The most commonly used failure definition is simple: failure occurs when the fault reaches a predetermined level. Prognosis performs the important linkage of the diagnostic information with the maintenance scheduler. It is probably the least understood but most crucial part of the diagnostic/prognostic/CBM hierarchical architecture. Furthermore, it entails ambiguity and large-grain uncertainty. For example, it is difficult to model the degradation of a failure accurately due to the variations of operating conditions and environmental effects, or historical data is not readily available.

5.2 Multiple Failure Prognosis by Particle Swarm Optimization

Figure 5.1: Degradation profile

In this work, the degradation of component is divided into two categories: parametric degradation and nonparametric degradation. Parametric degradation is characterized by the slowly changing of component parameter. For example, blockage of pipe is a typical example of incipient fault and the blockage, often modeled as resistance, usually increases over time. Nonparametric degradation refers to the deterioration of system components that are under monitoring but cannot be described by physical parameters. Examples include sensors, actuators, and subsystems.

Figure 5.1(a) shows the parametric degradation profile, in which curve A represents the gradual increasing of parameter due to parametric fault. For example, resistance value increases caused by pipe blockage. The curve B represents the gradual decreasing of parameter due to parametric fault. Figure 5.1(b) represents the nonparametric degradation profile, in which the efficiency factor β in (4.1) decreases gradually over time. In Figure 5.1, t_0 is the time point at which the degradation first starts. Ideally, deviations in residuals caused by faults and degradation should be detected at or very soon after fault occurrence. In practice, a certain delay between the time of fault occurrence and detection of fault can happen due to the fact that the degradation can occur at the non-detectable mode and is detected at a detectable mode, i.e. $t_d > t_0$. For example, in a hybrid system with $GARR_1 = V_1 - a_1 \cdot R_1 \cdot Df - De_1$, resistor R_1 degrades at a non-detectable mode, i.e. $a_1 = 0$, at t_0 . Since term $a_1 \cdot R_1 \cdot Df$ takes value 0 in $GARR_1$ at that mode, no matter what value R_1 takes, $a_1 \cdot R_1 \cdot Df$ is always equal to 0, $GARR_1$ is not sensitive to the fault in R_1 at $a_1 = 0$, so the fault cannot be detected by $GARR_1$. When the system enters a new mode, i.e. $a_1 = 1$, at t_d , the fault in R_1 can be detected by $GARR_1$.

A component can show different degradation behaviors due to different operation conditions [64]. For example, as a critical component of rotating machinery, bearings may exhibit different kinds of incipient faults, such as corrosion, spalling and cracking, due to different operating conditions including load and environments. The fault progression of different incipient faults may be linear or nonlinear. In order to accurately predict the progression of incipient fault, the following first/second order Ordinary Differential Equations (ODE) are used to describe the degradation progression of incipient faults

$$\begin{aligned}
\dot{F} &= a_1 \cdot F \\
\dot{F} &= a_2 \cdot F^2 \\
\ddot{F} &= a_3 \cdot \dot{F} + F + b_1 \\
\ddot{F} &= a_4 \cdot \dot{F}^2 + b_2 \cdot F + c_1
\end{aligned} \tag{5.2}$$

where F represents the fault component value, i.e. fault parameter value for parametric fault or β value for nonparametric fault; $a_1, a_2, a_3, a_4, b_1, b_2$ and c_1 are degradation model coefficients.

Adams *et al.* [128] used dynamic degradation models with an additive term which varies over time. In this paper, some prescribed models in (5.2) are deployed for the behaviors of parameters and/or β evolving over time. These four degradation models only serve as examples to illustrate the proposed methodology and these prescribed degradation models should be chosen based on applications and understandings of the faulty components.

Let's consider a hybrid system with m AGARRs $\bar{g}_l(\theta, \beta, n)$ for $l=1, 2, \dots, m$, where n denotes the discrete sampling index. Once a fault is detected, N samples' data are captured. Under no fault condition, $\theta = \bar{\theta}$ and $\beta = \bar{\beta}$, with $\bar{\theta}$ and $\bar{\beta}$ being the nominal values of parameters and efficiency factors. If a fault-detectable component, i.e., $\theta_i \in \theta$ or $\beta_i \in \beta$, is faulty, the AGARRs $\bar{g}_l(\bar{\theta}, \bar{\beta}, n)$ that contains θ_i and/or β_i will be nonzero, and thus a fault is detected. After a fault is detected, the fault isolation module is activated to find a set of suspected fault candidates. If a fault is detected at the time of mode change, then the fault candidates include those parameters θ_i and/or β_i of such fault signature which is non-detectable in the previous mode. This indicates that if a fault occurs at the detectable mode, it will be detected before mode changes; on the contrary, if a fault starts at the non-detectable mode, this fault will be detected only when the system enters a new mode in which the fault is detectable.

Consider Mode Dependent-FSM (MD-FSM) shown in Table 5.1 and Table 5.2. In the tables, $\alpha \in \{0, 1\}$ represents the operating mode of the hybrid system; θ_i , $i=1, \dots, 4$, denote the parameters; β_i , $i=1, \dots, 4$, represent the efficiency factors related to sensors or actuators; r_i , $i=1, \dots, 3$, denote the three residuals. The initial mode is

$a = 0$, and the mode $a = 1$. From Table 5.2, the fault could be due to changes of either θ_2 , θ_3 , β_1 , or their combination. It is noted that the fault is detected at the time of mode change, and only θ_3 and β_1 are non-detectable at previous mode $a = 0$. Therefore, the suspected fault candidates set could be $\delta = \{\theta_3, \beta_1, \theta_3 \& \beta_1\}$.

Table 5.1: MD-FSM at $a = 1$

	r_1	r_2	r_3	D_b
θ_1	1	0	0	1
β_1	0	0	0	0
θ_2	0	0	1	1
β_2	0	1	0	1
θ_3	0	0	0	0
β_3	0	1	0	1
θ_4	0	1	1	1
β_4	0	1	0	1

Table 5.2: MD-FSM at $a = 0$

	r_1	r_2	r_3	D_b
θ_1	1	0	0	1
β_1	0	0	1	1
θ_2	0	0	1	1
β_2	0	1	0	1
θ_3	0	0	1	1
β_3	0	1	0	1
θ_4	0	1	1	1
β_4	0	1	0	1

In the MD-FSM under multiple fault condition, the fault isolability (I_b) is not required. Unlike fault analysis under single fault assumption, the uniqueness of fault signature of a component can be due to one or multiple faults. For example, if both r_2 and r_3 are abnormal at mode $a = 1$, it leads to a coherence vector $[0 \ 1 \ 1]$. This fault signature is unique in Table 5.2, and is isolable under single fault assumption. However, for a system having multiple faults, the fault signature could be due to change of a single parameter θ_4 or due to simultaneous changes in more than one parameters, e.g. θ_2 and β_2 , θ_3 and β_4 , etc.

If the parameters θ and efficiency factors β are equal to the nominal values, then $\bar{g}_l(\theta, \beta, n) = 0$, for $l = 1, 2, \dots, m$. However, the parameters θ and efficiency factors β will change values over time due to degradation process. The identification of fault degradation requires matching the progressing profile from collected data to one of the prescribed degradation models. The matching is carried out through identifications of model coefficients. Once the model coefficients are determined, the identified model can be used to predict the RUL. For example, if θ_i is the true fault and the degradation profile matches the model of the first order nonlinear ODE, then the

AGARR $\bar{g}_i(\theta, \beta, n)$ that contains θ_i can be reformulated as

$$\begin{aligned} & \bar{g}_i(sol, \theta_j, \beta, n), \\ & \text{with } sol = -1/(a_2 \cdot t - 1/\theta_i(0)) \\ & t = n \cdot t_s, \quad j = 1, \dots, m \text{ and } j \neq i \end{aligned} \quad (5.3)$$

where t_s is the sampling period. Note that in the solution, the initial condition $\theta_i(0)$ is not equal to the nominal value of θ_i . Since the fault initiates at the non-detectable mode, $\theta_i(0)$ is also an unknown coefficient.

With the identification process, an approximate model can be selected from a set of prescribed models. Here, a model selection scheme is developed as follows. A binary vector is defined as $\eta = [\eta_1 \quad \eta_2]$ and let the four binary values to associate with the four prescribed models given in (5.2), where $[\eta_1 \quad \eta_2] = [0 \quad 0]$ means that the degradation model fulfills the first equation in (5.2), $[\eta_1 \quad \eta_2] = [0 \quad 1]$ indicates that the degradation model matches the second equation in (5.2), $[\eta_1 \quad \eta_2] = [1 \quad 0]$ means that the degradation model fulfills the third equation in (5.2), and $[\eta_1 \quad \eta_2] = [1 \quad 1]$ indicates that the degradation model matches the last equation in (5.2). In this way, the model selection is represented by the binary selection in the identification process.

When a fault is detected and a set of suspected faults is isolated, the prognosis module is activated to predict the RUL of the faulty components. The first step of prognosis is to select a prescribed model and to identify the model coefficients. The next step is to predict the RUL of the faulty component(s) based on the identified degradation model. The identification aims to find continuous real numbers for model coefficients and discrete binary values for degradation models simultaneously. However, binary numbers representing different degradation behaviors are not involved in the fitness function, so traditional gradient based method cannot directly tackle this identification problem. In other words, gradient based optimization methods such as least square method and Gauss-Newton method, usually require that the objective function is differentiable with respect to the optimized parameters. In this work, the objective function to be minimized is not directly related to the binary vector η , thus no gradient information can be obtained regarding the binary numbers.

Therefore, gradient-free stochastic optimization algorithms could be efficient alternatives. Stochastic optimization algorithms include ant colony optimization, GA, PSO, and so on. PSO does not have complex genetic operation and has fewer control parameters in PSO. All these features make PSO simple and easy to implement. As a result, PSO has been widely used in many areas, such as training neural networks [130, 131], control system design [129] and power system optimization [132].

The AHPSO developed in Chapter 3 is adopted to carry out the identification task, where RPSO is used to find continuous real numbers for model coefficients and initial values for certain degradation behavior, and BPSO is utilized to search the discrete binary values for degradation models. Figure 5.2 depicts the particle structure of AHPSO algorithm. In the figure, s is the number of coefficients for certain degradation model.

Figure 5.2: Particle structure of AHPSO algorithm

Both RPSO and BPSO evolve simultaneously and are coupled through the common fitness function defined as follows

$$F_{fitness} = 1 / (\sum_{l=1}^m \sum_{n=1}^N |\bar{g}_l^n| + \varepsilon) \quad (5.4)$$

where ε is a small positive constant to avoid zero division during search process.

AHPSO is applicable to the condition in which an isolable fault is detected. For instance, let us consider the example shown in Table 5.1 and Table 5.2, if residual r_1 is abnormal at mode $a = 1$, it leads to a coherence vector $[1 \ 0 \ 0]$, and the suspected faults set could be $\delta = \{\theta_1\}$. Then AHPSO can be adopted to identify the degradation model and model coefficients for θ_1 . However, if the suspected faults set consists of more than one elements, for example, $\delta = \{\theta_3, \beta_1, \theta_3 \ \& \ \beta_1\}$, a Multiple AHPSO (MAHPSO) algorithm is developed for the case of multiple elements in the set of targeted faults. In MAHPSO, number of P AHPSO estimators run in parallel, and P is equal to the number of elements in the set of targeted faults. The fitness values of various AHPSO estimators addressing different elements in the set are compared with

each other to choose the degradation model and model coefficients. MAHPSO based prognosis method is shown in Figure 5.3. Moreover, if the element in the set of targeted faults includes more than one fault, for example,

Figure 5.3: Block diagram of the MAHPSO based prognosis method

θ_3 & β_1 in $\delta = \{\theta_3, \beta_1, \theta_3 \& \beta_1\}$, the particle structure of corresponding AHPSO estimator needs to be modified, as shown in Figure 5.4. In the figure, n is the number of faults in the element in the set of targeted faults.

Figure 5.4: Particle structure of AHPSO algorithm considering multiple faults

According to (5.2), if the failure thresholds are given for different faults, then these thresholds are substituted into the corresponding solutions and the time to failure from the moment that faults are detected can be calculated. For example, consider a set of suspected faults $\delta = \{\theta_3, \beta_1, \theta_3 \& \beta_1\}$, where θ_3 and β_1 are true faults. The degradation profile of θ_3 matches the model of the first order nonlinear ODE, i.e. $\dot{\theta}_3 = a_2 \cdot \theta_3^2$ and

the degradation profile of β_1 matches the model of the first order linear ODE, i.e. $\dot{\beta}_1 = a_1 \cdot \beta_1$. With the identified faults, estimated degradation models and coefficients, the RUL for θ_3 can be calculated as $(\theta_{3F} - \theta_3(0)) / (\theta_{3F} \cdot \theta_3(0) \cdot a_2) - N \cdot ts$ and the RUL for β_1 can be computed as $\log(\beta_{1F} / \beta_1(0)) / a_1 - N \cdot ts$, where θ_{3F} and β_{1F} denote the failure thresholds of θ_3 and β_1 respectively.

5.3 An Illustrative Example

5.3.1 System Description

A hybrid system in electrical domain is considered as shown in Figure 5.5, whose DHBG is depicted in Figure 5.6. The system consists of five R elements $\{R_1, R_2, R_3, R_4, R_5\}$, two C elements $\{C_1, C_2\}$, two switches sw_1 and sw_2 , one current (flow) sensor Df , and three voltage (effort) sensors $\{De_1, De_2, De_3\}$. To derive AGARRs from the DHBG, storage elements are assigned with derivative causality, and sensor causality is inversed.

Figure 5.5: A hybrid system: an electric circuit

Figure 5.6: DHBG of the electric circuit

First, consider the constitutive relation of junction 1₁

$$e_1 - e_2 - e_3 = 0 \quad (5.5)$$

Since f_2 is measurable, $f_2 = Df$, and the causal paths lead e_2 to

$$e_2 = R_1 \cdot Df \quad (5.6)$$

Substituting (5.6) into (5.5), AGARR₁ is achieved

$$AGARR_1 = a_1(\beta_{V_1} \cdot V_1 - R_1 \cdot \frac{Df}{\beta_{Df}} - \frac{De_1}{\beta_{De_1}}) = 0 \quad (5.7)$$

Next, consider junction 0₂ attached to De_1 , the constitutive relation of this junction can be expressed as

$$a_1 f_3 + f_4 - f_7 - f_8 = 0 \quad (5.8)$$

Then tracking back the causal paths yields

$$\begin{aligned} f_3 &= Df \\ f_4 = f_5 &= \frac{1}{R_2} e_5 = \frac{1}{R_2} (e_6 - e_4) = \frac{1}{R_2} (V_2 - De_1) \\ f_7 &= C_1 \frac{d}{dt} e_7 = C_1 \frac{d}{dt} De_1 \\ f_8 = f_9 &= \frac{1}{R_3} e_9 = \frac{1}{R_3} (e_8 - e_{10}) = \frac{1}{R_3} (De_1 - De_2) \end{aligned} \quad (5.9)$$

Substituting (5.9) into (5.8) gives

$$AGARR_2 = a_1 \frac{Df}{\beta_{Df}} + \frac{1}{R_2} (\beta_{V_2} V_2 - \frac{De_1}{\beta_{De_1}}) - C_1 \frac{d}{dt} (\frac{De_1}{\beta_{De_1}}) - \frac{1}{R_3} (\frac{De_1}{\beta_{De_1}} - \frac{De_2}{\beta_{De_2}}) = 0 \quad (5.10)$$

Using junction 0₅ with De_2 , yields

$$f_{10} - f_{11} - a_2 f_{12} = 0 \quad (5.11)$$

By following the causal paths, the flow variables f_{10} , f_{11} and f_{12} can be represented as

$$\begin{aligned} f_{10} = f_9 &= \frac{1}{R_3} e_9 = \frac{1}{R_3} (e_8 - e_{10}) = \frac{1}{R_3} (De_1 - De_2) \\ f_{11} &= C_2 \frac{d}{dt} e_{11} = C_2 \frac{d}{dt} De_2 \\ f_{12} = f_{13} &= \frac{1}{R_4} e_{13} = \frac{1}{R_4} (e_{12} - e_{14}) = \frac{1}{R_4} (De_2 - De_3) \end{aligned} \quad (5.12)$$

The third AGARR₃ can be obtained by combining (5.11) and (5.12)

$$AGARR_3 = \frac{1}{R_3} \left(\frac{De_1}{\beta_{De_1}} - \frac{De_2}{\beta_{De_2}} \right) - C_2 \frac{d}{dt} \left(\frac{De_2}{\beta_{De_2}} \right) - a_2 \frac{1}{R_4} \left(\frac{De_2}{\beta_{De_2}} - \frac{De_3}{\beta_{De_3}} \right) = 0 \quad (5.13)$$

Finally consider junction 0₇, with the constitutive relation

$$f_{14} - f_{15} = 0 \quad (5.14)$$

Tracking the casual paths, gives

$$f_{14} = f_{13} = \frac{1}{R_4} e_{13} = \frac{1}{R_4} (e_{12} - e_{14}) = \frac{1}{R_4} (De_2 - De_3) \quad (5.15)$$

$$f_{15} = \frac{1}{R_5} e_{15} = \frac{1}{R_5} De_3 \quad (5.16)$$

Then $AGARR_4$ is given by

$$AGARR_4 = a_2 \left[\frac{1}{R_4} \left(\frac{De_2}{\beta_{De_2}} - \frac{De_3}{\beta_{De_3}} \right) - \frac{1}{R_5} \frac{De_3}{\beta_{De_3}} \right] = 0 \quad (5.17)$$

From the four $AGARR$ s derived, the MD-FSM can be obtained in Table 5.3~ Table 5.6. Note that β is used to represent the extent of the nonparametric fault, then β always has the same fault signature as that of its counterpart, e.g. β_{V_1} and V_1 . Consequently, in the Tables, source elements and sensors are replaced by their corresponding efficiency factors.

Table 5.3: MD-FSM at $a_1 = 0$, $a_2 = 1$ Table 5.4: MD-FSM at $a_1 = 1$, $a_2 = 0$

	r_1	r_2	r_3	r_4	D_b
β_{V_1}	0	0	0	0	0
β_{V_2}	0	1	0	0	1
C_1	0	1	0	0	1
C_2	0	0	1	0	1
R_1	0	0	0	0	0
R_2	0	1	0	0	1
R_3	0	1	1	0	1
R_4	0	0	1	1	1
R_5	0	0	0	1	1
β_{Df}	0	0	0	0	0
β_{De_1}	0	1	1	0	1
β_{De_2}	0	1	1	1	1
β_{De_3}	0	0	1	1	1

	r_1	r_2	r_3	r_4	D_b
β_{V_1}	1	0	0	0	1
β_{V_2}	0	1	0	0	1
C_1	0	1	0	0	1
C_2	0	0	1	0	1
R_1	1	0	0	0	1
R_2	0	1	0	0	1
R_3	0	1	1	0	1
R_4	0	0	0	0	0
R_5	0	0	0	0	0
β_{Df}	1	1	0	0	1
β_{De_1}	1	1	1	0	1
β_{De_2}	0	1	1	0	1
β_{De_3}	0	0	0	0	0

Table 5.5: MD-FSM at $a_1 = 0$, $a_2 = 0$

	r_1	r_2	r_3	r_4	D_b
β_{V_1}	0	0	0	0	0
β_{V_2}	0	1	0	0	1
C_1	0	1	0	0	1
C_2	0	0	1	0	1
R_1	0	0	0	0	0
R_2	0	1	0	0	1
R_3	0	1	1	0	1
R_4	0	0	0	0	0
R_5	0	0	0	0	0
β_{Df}	0	0	0	0	0
β_{De_1}	0	1	1	0	1
β_{De_2}	0	1	1	0	1
β_{De_3}	0	0	0	0	0

Table 5.6: MD-FSM at $a_1 = 1$, $a_2 = 1$

	r_1	r_2	r_3	r_4	D_b
β_{V_1}	1	0	0	0	1
β_{V_2}	0	1	0	0	1
C_1	0	1	0	0	1
C_2	0	0	1	0	1
R_1	1	0	0	0	1
R_2	0	1	0	0	1
R_3	0	1	1	0	1
R_4	0	0	1	1	1
R_5	0	0	0	1	1
β_{Df}	1	1	0	0	1
β_{De_1}	1	1	1	0	1
β_{De_2}	0	1	1	1	1
β_{De_3}	0	0	1	1	1

5.3.2 Simulation Results

(a) Degradation profile in R_4 (b) Degradation profile in β_{De_3}

Figure 5.7: Degradation profile in electrical circuit

The physical parameters of the circuit are: $V_1 = 7\text{V}$, $V_2 = 1\text{V}$, $R_1 = 670\Omega$, $R_2 = 215.6\Omega$, $R_3 = 67.5\Omega$, $R_4 = 215.4\Omega$, $R_5 = 509\Omega$, $C_1 = 10000\mu\text{F}$ and $C_2 = 4700\mu\text{F}$, sampling time t_s is 0.05 s , $\varepsilon = 0.01$ in (5.4). Two incipient faults are introduced. The first one is a gradual buildup of R_4 at $t = 5\text{ s}$, the degradation profile matches the model of the first order linear ODE in (5.2) with coefficients $a_1 = 0.04$,

failure value for R_4 is chosen as $R_{4F} = 800\Omega$. The other one is a gradual degradation in sensor De_3 introduced in β_{De_3} at $t = 6s$, the degradation profile fulfills the model of the first order nonlinear ODE in (5.2) with coefficients $a_2 = -0.03$, failure value for β_{De_3} is set as $\beta_{De_3F} = 0.4$. In simulation, the sensor fault in De_3 is realized by multiplying the normal sensor reading by β_{De_3} in Simulink, where β_{De_3} is modeled as a function of time using function block in Simulink library. The degradation profile is shown in Figure 5.7. Note that in real industrial applications it is commonplace for the fault to evolve slowly over a period of months, sometimes even over years. In order to avoid excessively long period simulations, the fault development rate has been increased so that significant effects are present after a few seconds [133]. The switches programming is shown in Figure 5.8. Figure 5.9 presents the simulation responses of residuals due to the faults in R_4 and β_{De_3} , in the figure, dash lines denote the thresholds $\varepsilon_1 = 1e-6$, $\varepsilon_2 = 0.03$, $\varepsilon_3 = 0.6e-3$ and $\varepsilon_4 = 5e-4$. These thresholds are not calculated, they are set by observing the normal residual responses. If the residual exceeds the predetermined threshold, then system is considered as faulty. According to the MD-FSM tables, it is known that the faults in R_4 and β_{De_3} initiate at a non-detectable mode, i.e. $a_1 = 0$ and $a_2 = 0$. The faults cannot be detected until the system enters a mode in which the faults are detectable, i.e. $a_1 = 1$ and $a_2 = 1$. Figure 5.9 reveals that residual r_3 and r_4 are sensitive to the faults at mode $a_1 = 1$ and $a_2 = 1$. From the MD-FSM table at that mode, R_4 and β_{De_3} have fault signature $[0 \ 0 \ 1 \ 1]$ which matches the simulation result. All observations have been captured from the moment that a fault is detected. With an observation window of 10 seconds, two

Figure 5.8: Switches programming for simulation

Figure 5.9: Simulation residual responses for resistance fault and sensor fault

hundreds of sample data ($N = 200$) are collected. According to the MD-FSM table at mode $a_1 = 1$ and $a_2 = 1$, R_4 , R_5 and β_{De_3} are non-detectable at previous mode $a_1 = 1$ and $a_2 = 0$. The coherence vector $[0 \ 0 \ 1 \ 1]$ may result from one or more faults occurring simultaneously, and the faults are detected at the time of mode change, hence the targeted faults set is

$$\delta = \{R_4 \& R_5 \& \beta_{De_3}, R_4 \& R_5, R_4, \beta_{De_3}, R_5 \& \beta_{De_3}, R_4 \& \beta_{De_3}\}.$$

The failure prognosis module is triggered and six AHPSO estimators in MAHPSO run in parallel, each based on one element from the targeted faults set. The parameters associated with AHPSO are taken as: Particle number=350, Maximum iterations=400, $\Psi_1 = 2$, $\Psi_2 = 2$, $w_r = 0.5$, $V_{\max, r} = 4$. In order to eliminate the influence of randomness, each AHPSO is carried out 20 times from different initial seeds of the random number to ensure the repetitiveness of convergence. It has been observed that the final solutions obtained with different parameter settings do not differ much (standard deviation is less than 2% of the mean value). Consequently, results presented are the mean of the 20 times. Figure 5.10 shows the evolution of mean fitness value versus iterations. It is obvious that the faults are in $R_4 \& \beta_{De_3}$, not in the other elements in the targeted faults set.

Figure 5.10: Mean fitness evolution versus iterations for simulation

Table 5.7 summarizes the prognosis results, in which $R_4(0)$ and $\beta_{De_3}(0)$ are the initial values of R_4 and β_{De_3} in the collected data. The identified degradation model for R_4 is the first order linear ODE in (5.2) according to the binary vector $[\eta_1 \ \eta_2] = [0 \ 0]$ which matches the designed value. Similarly, binary vector $[\eta_1 \ \eta_2] = [0 \ 1]$ for β_{De_3} indicates that its degradation process fulfills the second equation in (5.2) and the result matches the designed value. The estimated RUL is 17.88s for R_4 and 36.3s for β_{De_3} , which are close to the designed values 17.85s and 36.1s. The simulation results show that the proposed prognosis method can accurately track the degradation dynamics and predict the RUL for hybrid systems.

Table 5.7: Summary for simulation results

Fault in R_4	a_1	$R_4(0)$	η_1	η_2	RUL
Designed values	0.04	263.09	0	0	17.85s
Estimated values	0.041	262.46	0	0	17.88s
Fault in β_{De_3}	a_2	$\beta_{De_3}(0)$	η_1	η_2	RUL
Designed values	-0.03	0.893	0	1	36.1s
Estimated values	-0.032	0.894	0	1	36.3s

5.3.3 Experiment Results

Figure 5.11: An experimental setup for FDI and prognosis

As shown in Figure 5.11, a circuit test-bed is established to verify the proposed method. The real time FDI is based on the MATLAB Real-Time Windows Target, and two data acquisition cards (NI PCI-6025E and NI PCI-6713). The switches are implemented by two PhotoMOS relays (AQV251) and controlled by the real time software. The physical parameters of circuit and sampling time are same as those used in simulation. Due to the numerical differentiation of noisy measurements in the AGARRs, the results for estimated variables are noisy. To get good estimates, a low pass filter is deployed to filter out the effect of measurement noise on residuals.

Figure 5.12: Degradation profile in β_{V_1}

Figure 5.13: Switches programming for experiment

An actuator fault occurs when there is discrepancy between command input of an actuator and its actual output to system. It is introduced by reducing β_{V_1} at moment $t = 5s$ as shown in Figure 5.12. The degradation profile of β_{V_1} matches the model of the second order linear ODE in (5.2), the failure value for β_{V_1} is chosen as $\beta_{V_1F} = 0.5$. In the experiment, the actuator fault is realized by multiplying input V_1 by β_{V_1} in

Simulink under Real-Time Windows Target environment, where β_{V_1} is modeled as a function of time using function block in Simulink library and unknown for the user. The switches programming is shown in Figure 5.13. Since the fault is injected at mode $a_1 = 0$ and $a_2 = 0$, in which β_{V_1} is non-detectable, so the fault cannot be detected by the residuals as shown in Figure 5.14. Until $t = 10$ s the system enters a new mode $a_1 = 1$ and $a_2 = 1$, from MD-FSM tables, the fault signature of β_{V_1} is $[1 \ 0 \ 0 \ 0]$ which indicates it is detectable. However, the fault in β_{V_1} cannot be isolated because R_1 shares the same fault signature with the one of β_{V_1} , and both β_{V_1} and R_1 are non-detectable at previous mode $a_1 = 0$ and $a_2 = 1$, thus $\delta = \{R_1, \beta_{V_1}, R_1 \& \beta_{V_1}\}$. Finally, the failure prognosis module is invoked in which three AHPSO estimators in MAHPSO run in parallel. Figure 5.15 depicts the evolution of mean fitness value versus iterations. It is obvious that the degradation is in β_{V_1} , not in R_1 and $R_1 \& \beta_{V_1}$. The prognosis results are listed in Table 5.8. The identified degradation model for R_1 is the second order linear ODE in (5.2) according to the

Figure 5.14: Experimental residual responses for actuator fault

Figure 5.15: Mean fitness evolution versus iterations for experiment

Table 5.8: Summary for experiment results

Fault in β_{v_1}	a_3	b_1	$\beta_{v_1}(0)$	η_1	η_2	RUL
Designed values	-50	-1.2	0.979	1	0	47.7s
Estimated values	-48.84	-1.19	0.972	1	0	47.3s

binary vector $[\eta_1 \ \eta_2] = [1 \ 0]$ which matches the designed value. The results confirm that the MAHPSO based prognosis method performs well in real experiment.

5.3.4 Comparison Study

For comparison, AHPSO, GA and Adaptive GA (AGA) [107] approaches have been tested on the same experimental data collected, since the true fault is β_{v_1} , then GA and AGA methods are carried out considering β_{v_1} as fault. The parameters of GA are set as: Population=350, Maximum iterations=400, $p_c = 0.6$, $p_m = 0.04$. The parameters of AGA are the same as those of GA, the adaptive scheme of AGA consists of some rules to adaptively tune the crossover and mutation rates according to the performance of the current genetic operators. It tries to increase the probability of the genetic operator if it consistently produces a better offspring during the search

process; on contrast, it tries to reduce the probability of the genetic operator if it always produces a poorer offspring. This method can adaptively regulate the balance between the exploration and exploitation of the solution space.

Comparison results are shown in Table 5.9 in which all methods identify that the degradation model for R_1 is the second order linear ODE in (5.2) according to the binary vector $[\eta_1 \ \eta_2] = [1 \ 0]$. It is observed that AHPSO is superior to the other two methods because AHPSO has better information sharing and conveying mechanism than GA and AGA. AHPSO also has good dynamics of balance between global and local search abilities, and it can easily adjust inertia weight of RPSO and maximum velocity of BPSO using simple “IF–THEN” rules.

Table 5.9: Comparison results using different methods

Methods	a_3	b_1	$\beta_{V_1}(0)$	η_1	η_2	RUL
AHPSO	-48.84	-1.19	0.972	1	0	47.3s
GA	-47.93	-1.12	0.961	1	0	48.5s
AGA	-48.56	-1.17	0.969	1	0	47.1s

5.4 Conclusions

The novel scheme proposed for multiple failures prognosis for hybrid systems based on MAHPSO has been verified by simulation as well as experiments. The fundamental contributions in this chapter include the utilization of the AGARRs to describe the system behaviors in the presence of incipient faults of parametric and nonparametric nature; multiple incipient faults analysis extends the application scope of model-based health monitoring framework; the degradation models which enable the failure prognosis; as well as the application of AHPSO for fault and degradation model identification. The scheme is now able to carry out the failure prognosis if the faults occur at a non-detectable mode and there is a transition of modes from non-detectable to detectable.

Chapter 6

Online Fault Detection and Isolation in A Mobile Robot Test Bed

The ultimate goal of the fault diagnosis algorithms is to solve real problems. Diagnosis approaches should be tested and evaluated on real world systems to exhibit practical significance. In this chapter, the Quantitative Hybrid Bond Graph (QHBG) based FDI framework is studied in details using a complex mechatronic system – a CyCab mobile robot test-bed.

6.1 The CyCab Mobile Robot Test Bed

Figure 6.1: The steering system of CyCab mobile robot

The CyCab mobile robot under consideration consists of four driving wheels and dual steering system, and the steering system of the mobile robot includes DC motor, reducing gear, belt, oil pump, hydraulic piston, steering mechanism and tires as shown in Figure 6.1. The DC motor speed is measured by an incremental encoder mounted on the reducing output shaft. A piston inside the hydraulic actuator is connected to the

Ackerman steering mechanism which drives the wheels to turn. The steering angle is measured by an absolute encoder installed beneath the centre platform. There are two pressure sensors, each installed on one side of the cylinder, are utilized to measure the oil pressure inside the cylinder as depicted in Figure 6.2.

Figure 6.2: Piston internal leakage simulated by a valve

Seven different faults, representing sensor faults, parametric faults and fault modes are considered. Each one of the four sensors in the steering system is a possible source of fault. Two fault modes are considered, the first is a burnt DC-motor or a burnt-driver; in this case no power is delivered to the steering system. The second fault mode is a broken-belt, where the motor and gear are rotating but power is not delivered to the pump.

Two parametric faults are discussed, a flat tire and a piston internal leakage. Flat tire fault is detected if the parameters representing the friction at the contact surface between the tire and the road are not consistent with actual system behavior. In order to simulate this fault, the valve of the wheel is opened and the air is gradually removed from the wheel. The flat tire fault is an incipient parametric fault. Internal leakage in an actuating cylinder (piston), due to worn seal, is a common fault in hydraulic machines [134]. The piston divides the cylinder into two chambers, which are filled with oil. When the pump rotates in one direction, oil pressure is generated in one of the two chambers; this is the high pressure side. The pressure at the other side

is low. The oil at the high pressure side pushes the piston and the piston drives the steering wheels through a steering mechanism. The oil at the low pressure side returns to the oil reservoir at the pump, through the low pressure tube. For normal operation of hydraulic actuator, oil is not supposed to flow from one chamber to the other. If the internal flow between the chambers is not negligible, the piston efficiency is decreased. The piston internal leakage fault is illustrated at the bottom part of Figure 6.2.

To simulate the internal leakage fault, an oil bypass has been installed, to allow free oil flow from one side to the other side of the cylinder. A proportional (manual) valve is used to control the oil flow in the bypass tube. When the valve is completely closed, it represents the normal condition. An open valve represents abnormal condition, and the fault severity is determined by the opening level of the valve. The oil piston, cylinder, bypass tubes and the valve are presented in Figure 6.2 (the valve is marked by a circle).

6.2 HBG Model of CyCab Electro-hydraulic Steering System

Figure 6.3: DHBG model of the front steering system

The DHBG model of the front steering system is demonstrated in Figure 6.3. u_{in} represents the input signal to the steering system, which is modeled by source element Ms_f . The torque generated by the motor is proportional to the current. The voltage to current ratio is k_1 , and the current to torque ratio is k_2 . The motor inertia is modeled by J_1 , and the friction is modeled by R_1 , which includes the viscous friction

coefficient k_f and coulomb friction coefficient F_u . The relation between effort and flow, due to the mechanical friction in motor, is described as

$$e = k_f \cdot f + F_u \text{sign}(f) \quad (6.1)$$

The binary state variable a_1 in Figure 6.3 represents a burnt-driver and/or a burnt-motor fault. In case of fault, i.e. $a_1 = 0$, the current through the motor is equal to zero. $a_1 = 1$ indicates the normal operation condition. The reducing gear ratio is k_3 , and belt is modeled by a TF with a transmission ratio k_4 .

An incremental encoder measures the shaft velocity $\dot{\theta}_1$ at the gear output. The state variable a_2 is utilized to model the broken belt fault. The faulty state is represented by $a_2 = 0$, which disconnects the pump from the motor. Note that the pump velocity is zero if motor torque is not delivered to the pump. For the hydraulic pump, it consists of rotor inertial J_2 and rotor friction R_2 , including k_{f_1} and F_{u_1} . k_5 represents the ratio between the pump angular velocity and pump oil flow. The pump is bidirectional and equipped with two internal pressure release valves. These are not given in the model and have no influence on the results presented in this work. Two oil tubes connect the pump with an oil piston; assume the oil compressibility and tubes elasticity are negligible in this study.

The piston is modeled by a TF with a ratio A , which represent the two effective cross section area of the piston. The pressure difference p between two pressure sensors is modeled by De . The piston internal leakage is modeled by R_l which takes value $R_l \rightarrow \infty$ under normal condition and $R_l \ll \infty$ under leakage condition. The relation between pressure difference p and oil flow through a valve or through a leak is described as

$$f = \left(\text{sign}(p) \sqrt{|p|} \right) / R_l \quad (6.2)$$

The steering mechanism is modeled by $F_1(\theta_2)$ and $F_2(\theta_2)$ which are derived from the Ackerman's geometry as shown in Figure 6.4. The goal of this mechanism is to turn the two steering wheels in different steering angles, so the tire slippage is minimized. For modeling purpose, the two steering wheels are represented by a single virtual wheel, located at the center point between the two actual wheels.

Figure 6.4: The Ackerman steering mechanism

The steering angle of the virtual wheel is γ , and is given as

$$\gamma = \frac{\gamma_{in} + \gamma_{out}}{2}, \quad \dot{x} = F_1(\theta_2)\dot{\theta}_2, \quad \dot{\gamma} = F_2(\theta_2)\dot{\theta}_2 \quad (6.3)$$

where x is the piston position.

The steering angle is measured by an absolute encoder, located at point D in Figure 6.4. The cylinder is fixed to the mobile robot frame at point G . The inertia of the virtual steering wheel is J_3 and friction between road and tire R_3 , including k_{f_2} and Fu_2 .

First, consider the constitutive relation of junction 1₃

$$e_{11} = e_{13} \quad (6.4)$$

Then tracking back the causal paths yields

$$\begin{aligned} e_{11} &= \frac{1}{k_3} e_{10} = \frac{1}{k_3} (e_7 - e_8 - e_9) = \frac{1}{k_3} [k_2 f_6 - J_1 \dot{f}_8 - k_f f_9 - F_u \text{sign}(f_9)] \\ &= \frac{1}{k_3} \left[a_1 k_2 k_1 u_{in} - J_1 \frac{1}{k_3} \ddot{\theta}_1 - k_f \frac{1}{k_3} \dot{\theta}_1 - F_u \text{sign}(\dot{\theta}_1) \right] \end{aligned} \quad (6.5)$$

$$\begin{aligned} e_{13} &= a_2 k_4 e_{15} = a_2 k_4 (e_{14} + e_{15} + e_{16}) = a_2 k_4 [J_2 \dot{f}_{14} + k_{f_1} f_{15} + F_{u_1} \text{sign}(f_{15}) + k_5 e_{17}] \\ &= a_2 k_4 [J_2 k_4 \ddot{\theta}_1 + k_{f_1} k_4 \dot{\theta}_1 + F_{u_1} \text{sign}(\dot{\theta}_1) + k_5 p] \end{aligned} \quad (6.6)$$

Substituting (6.6) and (6.5) into (6.4) gives

$$\begin{aligned} GARR_1 &= \frac{a_1}{k_3} k_1 k_2 u_{in} - \left(\frac{J_1}{k_3^2} + a_2 J_2 k_4^2 \right) \ddot{\theta}_1 - \left(\frac{k_f}{k_3^2} + a_2 k_{f_1} k_4^2 \right) \dot{\theta}_1 \\ &\quad - \left(F_u \frac{1}{k_3} + a_2 k_4 F_{u_1} \right) \text{sign}(\dot{\theta}_1) - a_2 k_4 k_5 p = 0 \end{aligned} \quad (6.7)$$

Next, consider 0_1 junction; the constitutive relation of this junction can be expressed as

$$f_{19} - f_{21} - f_{22} = f_{20} = 0 \quad (6.8)$$

By following the causal paths, the flow variables f_{19} and f_{22} can be represented as

$$f_{19} = k_5 f_{18} = k_4 k_5 \dot{\theta}_1 \quad (6.9)$$

$$f_{21} = \left(\text{sign}(p) \sqrt{|p|} \right) / R_l \quad (6.10)$$

$$f_{22} = A_1 f_{23} = A_1 \frac{1}{F_1} \dot{\theta}_2 \quad (6.11)$$

According to (6.8) ~ (6.11), the second GARR can be written as

$$GARR_2 = a_2 k_4 k_5 \dot{\theta}_1 - \left(\text{sign}(p) \sqrt{|p|} \right) / R_l - A_1 \frac{1}{F_1} \dot{\theta}_2 = 0 \quad (6.12)$$

Considering the junction 1_5 yields

$$e_{24} - e_{26} = e_{25} = 0 \quad (6.13)$$

$$e_{24} = \frac{1}{F_1} e_{23} = \frac{1}{F_1} A_1 e_{22} = \frac{1}{F_1} A_1 D e \quad (6.14)$$

$$\begin{aligned} e_{26} = e_{27} F_2 = F_2 (e_{28} + e_{29}) &= F_2 \left[J_3 \dot{f}_{28} + k_{f_2} f_{29} + F_{u_2} \text{sign}(f_{29}) \right] \\ &= F_2 J_3 \frac{d}{dt} (F_2 \dot{\theta}_2) + k_{f_2} F_2^2 \dot{\theta}_2 + F_2 F_{u_2} \text{sign}(F_2 \dot{\theta}_2) \end{aligned} \quad (6.15)$$

The third GARR can be obtained by combining (6.13), (6.14) and (6.15)

$$GARR_3 = \frac{1}{F_1} A_1 p - J_3 F_2 \frac{d}{dt} (F_2 \dot{\theta}_2) - k_{f_2} F_2^2 \dot{\theta}_2 - F_{u_2} F_2 \text{sign}(F_2 \dot{\theta}_2) = 0 \quad (6.16)$$

6.3 FDI Approach for the Front Steering System

The FDI process of the electro-hydraulic steering system is based on the online evaluation of the three GARRs derived in (6.7), (6.12) and (6.16). In this work, only three modes are considered, i.e. $[a_1 \ a_2] = [1 \ 1]$ for the normal operation, $[a_1 \ a_2] = [1 \ 0]$ for the burnt driver and $[a_1 \ a_2] = [0 \ 1]$ for the broken belt. Since the friction between the tire and the road surface is relatively high, if the input u_{in} is below certain value, then the steering speed is around zero which is too low for

reliable FDI. FDI in the steering system is possible only if $u_{in} > u_{in_min}$. The steering wheel model does not include static friction, which might be dominant if the steering velocity is zero, i.e. a high oil pressure is required to start the steering motion. Consequently, $GARR_3$ is valid only if $\theta_2 > \theta_{2_min}$. In order to eliminate misleading information provided by GARRs in real experiment, the three GARRs are reformulated as:

$$\begin{aligned}\overline{GARR}_1 &= c_1 GARR_1 \\ \overline{GARR}_2 &= c_1 GARR_2 \\ \overline{GARR}_3 &= c_1 c_2 GARR_3\end{aligned}\tag{6.17}$$

with

$$c_1 = \begin{cases} 1 & \text{if } u_{in} > u_{in_min} \\ 0 & \text{else} \end{cases}, \quad c_2 = \begin{cases} 1 & \text{if } \dot{\theta}_2 > \dot{\theta}_{2_min} \\ 0 & \text{else} \end{cases}\tag{6.18}$$

In this work, the single fault assumption is adopted. This single fault could be a parametric fault or a fault mode. The mode $[a_1 \ a_2] = [1 \ 1]$ stands for the normal operation condition, and it is the only mode where a parametric fault could happen. Hence only the MD-FSM in mode $[a_1 \ a_2] = [1 \ 1]$ as shown in Table 6.1 is required for parametric fault isolation. In the table, R_l represents the internal leakage fault of piston. The friction coefficients $R_3 : \{k_{f_2} \ F_{u_2}\}$ stand for flat-tire fault. If the tire is flat, the friction is higher than normal which varies the values of the friction coefficients. $\dot{\theta}_1$, $\dot{\theta}_2$ and p represent the incremental sensor fault, absolute sensor fault and pressure fault, respectively. Note that although p appears in all three GARRs, not all of them are sensitive to this fault. When R_l is normal, i.e. $R_l \rightarrow \infty$, $GARR_2$ is insensitive to the pressure sensor fault. Under single fault assumption, if p is faulty then R_l should be normal which indicates that only $GARR_1$ and $GARR_3$ are sensitive to this fault. According to (6.17) and (6.18), parametric fault isolation is only possible when $c_1 = c_2 = 1$.

The MCSM is listed in Table 6.2. In the steering system, the initial mode is $[a_1 \ a_2] = [1 \ 1]$. The initial mode is fed into the three GARRs and as long as inconsistency is not detected by GARRs. If an inconsistency is detected by GARRs,

then the detected fault signature is utilized for mode change isolation, or for parametric-fault isolation. If the signature is $[1 \ 0 \ 0]$ or $[1 \ 1 \ 0]$, then a mode-change is hypothesized from Table 6.2, and a mode identification process is activated. In the mode identification process, ARR of suspected modes are derived from the GARRs and evaluated in real time. If all ARRs of a suspected mode are consistent with the monitored system, then new mode is identified. If new mode is not identified, then it is concluded that the system mode is $[a_1 \ a_2] = [1 \ 1]$ and the inconsistency is due to a parametric fault. The detected fault signature and the MD-FSM in Table 6.1 are used for parametric fault isolation.

Table 6.1: MD-FSM at mode $[a_1 \ a_2] = [1 \ 1]$

Mode $[a_1 \ a_2] = [1 \ 1]$	r_1	r_2	r_3	D_b	I_b
R_l	0	1	0	1	1
$R_3 : \{k_{f_2} \ F_{u_2}\}$	0	0	1	1	1
$\dot{\theta}_1$	0	1	1	1	1
$\dot{\theta}_2$	1	1	0	1	1
p	1	0	1	1	1

Table 6.2: The MCSM of the steering system

C-J	r_1	r_2	r_3	D_b	I_b
a_1	1	0	0	1	1
a_2	1	1	0	1	1

6.4 Experiment Study

6.4.1 Experimental Hardware and Software

Figure 6.5: Structure and hardware configuration of Cycab steering system

The mobile robot has four wheels which are driven by four independent DC motors. An incremental encoder is fitted on each DC motor to measure the wheel's angular velocity. An absolute encoder is installed on the robot's steering system to measure the steering angle. An incremental encoder mounted on the reducing output shaft of DC motor for steering is used to measure the angular velocity. Figure 6.5 depicts the overall structure and hardware configuration of Cycab steering system. A notebook with Window operating system hosts the online FDI software. The software uses MatlabR2007B with Simulink xPC Target to run and control real-time applications on the target PC. The host PC, i.e. the notebook, builds the FDI model using Simulink blocks.

The Microsoft Visual C++ compiler in host PC creates executable code according to the Simulink FDI model. The executable code is downloaded from the host PC to

the target PC running the xPC Target real-time kernel. After downloading the executable code, one can run and test the target application in real time.

There are two circuit boards, namely base board and upper board, for connecting vehicle's I/O signals and sensor data to xPC target's PCI cards. The main purpose of base board is to connect all the sensors and I/Os into one place, and make a system more manageable and easy to understand. Signals that connected to base board include, driving signal, steering signal, absolute encoder signal, incremental encoder signal and pressure sensor signals. The upper board is to connect the all signals from the base board to PCI devices.

The steering system is an open-loop system. The motion of the DC motor of the steering system is generated by a current-modulated PWM voltage from a DC motor amplifier. The translation of the piston is created under the pressure of the oil inside the cylinder, which in turn is pumped by a set of a DC motor and a hydraulic pump.

The target PC carries out the real time residual computation using measurements acquired from the sensors. All the measurements from the sensors are interfaced with the target PC via the PCI cards. The host PC performs the online FDI decision making using residual measurements acquired from the target PC. The host computer accesses the low-level target PC (onboard computer) through wireless communication using local wireless network.

Figure 6.6: Simulink model of data acquisition for the incremental encoder and absolute encoder signals

Figure 6.6 describes the Simulink model of data acquisition for the incremental encoder and absolute encoder signals. The signal from the absolute encoder is the steering angle. In order to obtain the steering angular velocity, the difference operation is used. Figure 6.7 depicts the Simulink model of data acquisition for pressure sensor signals. The NI PCI-6025E data acquisition card is used for this purpose. All of the signals obtained are fed into the FDI block as shown in Figure 6.8.

Figure 6.7: Simulink model of data acquisition for the pressure sensor signals

Figure 6.8: Simulink model of FDI process

Figure 6.9: Internal structure of Simulink model of FDI process

Figure 6.9 demonstrates the internal structure of Simulink model of FDI process. In the model, signals from the pressure sensors are voltage signals and 1 Volt is equivalent to 5 bar. Thus these signals need to be multiplied by 5 before they are fed into the GARR equations. For ease of analysis, the absolute values of the residuals are used for final decision making process.

The online FDI decision making is carried out using Matlab m-file. In real experiment, the residual threshold selection is set by trial and error to reduce the possibilities of misdetection and false alarm. It is not reliable to consider the system as faulty if only one or two sampling residual values exceed the threshold during observation. So if three or more successive residual values exceed the predefined threshold, the system is considered as faulty. After the fault occurrence, if three or more successive residual values return to threshold, then the system is considered as normal. Once a fault detected, the fault signature is recorded and

compared with the MD-FSM in Table 6.1 and the MCSM in Table 6.2 to carry out fault isolation.

6.4.2 Experiment Result

The physical parameters of the CyCab steering system are listed in Table 6.3. Some of these parameters are taken from hardware datasheet and others were identified by the GA technique. The input signal is defined as

$$V_{in} = 0.17(5 \sin 2t + \sin t + 0.5 \sin 0.5t + 0.2 \sin 2.5t) \quad (6.19)$$

Table 6.3: Physical parameters of the Cycab front steering system

Parameter	Value	Unit
J_1	3.727e-5	kgm ²
J_2	5.627e-4	kgm ²
J_3	9.78e-2	kgm ²
k_1	3	A/V
k_2	0.663	Nm/A
k_3	4/70	/
k_4	1/2	/
k_5	2.5942e-6	m ³
A	4.46e-4	m ²
k_f	1.1017e-5	Nm·s
F_u	6.20e-2	Nm
k_{f_1}	3.161e-1	Nm·s
F_{u_1}	3.55e-2	Nm
k_{f_2}	33.28	Nm·s
F_{u_2}	22.11	Nm

Three different fault scenarios are demonstrated here. The first one is a burnt-driver fault and this fault is modeled as a fault mode. During real system operation, the driver is disabled from $t = 30s$ to $t = 50s$. Figure 6.10 depicts the residual responses (absolute value) under burnt driver fault. The dash lines are thresholds whose values are listed in Table 6.4. The residual values are not zeros over some time interval due to numerical inaccuracies caused by derivative approximation in Simulink model and

noise signal in pressure sensor. It is observed that the detected signature is $[1 \ 0 \ 0]$ which matches the fault signature of a_1 . Therefore, a burnt driver fault is detected and isolated. The second fault scenario is a pressure sensor fault in p . Due to a shortcut in the sensor electrical circuit, the sensor outputs is 5 Volt (which is equivalent to 25bar, in a normal sensor). The fault is introduced around $t = 30s$, and the real time residual responses are shown in Figure 6.11. The detected signature of this fault is $[1 \ 0 \ 1]$. According to Table 6.1, a pressure sensor fault is detected and isolated. In the last fault scenario, the piston internal leakage is demonstrated. This fault is simulated by a dedicated valve, which has been installed in the hydraulic system. From $t = 23s$ to $t = 42s$, the valve is slightly opened; the piston is still working, but its efficiency is decreased. The residual responses are shown in Figure 6.12. The detected signature is $[0 \ 1 \ 0]$, a parametric fault in R_l is isolated according to Table 6.1.

Table 6.4: Thresholds for the mobile robot FDI

	Th_1	Th_2	Th_3
	1.1	$0.7e-5$	60

Figure 6.10: Real time residual responses for a fault mode- burnt driver fault in a_1

Figure 6.11: Real time residual responses for a sensor fault - pressure sensor fault in p

Figure 6.12: Real time residual responses for a
parametric fault—internal leakage in R_l

6.5 Conclusions

This chapter demonstrates an application of the QHBG based fault diagnosis framework for a complex hybrid system. The method is studied in details using an electro-hydraulic steering system of a CyCab mobile robot. The presented case study reveals the complexity and difficulties of hybrid system diagnosis, where fault modes, sensor faults and parametric faults are mixed in a single system. The structure and hardware configuration of CyCab steering system are discussed. The online FDI decision making is introduced. Three different fault scenarios are demonstrated and the real time experimental results show the efficiency of the method.

Chapter 7

Conclusions and Future Research

7.1 Conclusions

Health monitoring of hybrid systems has gained a lot of interest in the past few decades due to its various industrial application. The purpose of this work is to introduce a new perspective on fault diagnosis and prognosis for hybrid systems, which takes advantages of both bond graph modeling technique and computational intelligence. The proposed approaches are demonstrated and evaluated on several hybrid systems, such as electrical system, suspension system and mobile robot system. The main contributions of the thesis are summarized as follows:

The first achievement is the development of a technique to simultaneously identify the fault parameter value and mode change stamps. Since model based mode tracking technique is only useful before a fault is detected, the mode changes are parameterized as a function of IMC, switching time stamps and time to facilitate the identification task when the mode change occurs after fault occurrence. The utilization of gradient free stochastic optimization makes it possible to identify the unknown variables when no gradient information is available. These findings shed light on strategies for the development of efficient estimation scheme to achieve fault parameter identification in the presence of unknown mode changes. The proposed method is studied in two hybrid systems, namely an electrical circuit and electro-hydraulic suspension, to demonstrate the efficiency.

The second achievement is to propose a methodology to deal with multiple faults FDI and estimation without mode change and fault pattern information. An Augmented version of GARRs, called AGARRs, is proposed to describe the hybrid system with fault of parametric and non-parametric nature. Since in real application,

the fault under consideration could be abrupt or incipient fault, and the nature of the fault is unknown beforehand. It is difficult to carry out fault estimation if the fault pattern is unknown. The condition is further complicated by the presence of unknown mode changes. In order to reduce the computational complexity of the method, a binary vector is defined to represent different fault pattern for different suspected faults. A new multiple faults isolation scheme is introduced without fault pattern and mode change information. The faults may happen at both detectable mode and non-detectable mode. Different fault scenarios are studied to validate the proposed methodology.

The third achievement is the development of multiple failure prognosis for hybrid systems. The concept of AGARRs proposed in Chapter 4 is utilized to describe the incipient fault of parametric and non-parametric nature. The degradation models which enable the failure prognosis are developed to describe the degradation behaviors of the incipient faults. The degradation behavior of each incipient fault is unknown for the designer and a degradation model selection scheme is discussed. The developed prognosis method includes two steps: degradation model and degradation model coefficients selection, and prediction of the RUL(s) for faulty components. The faults may occur at a non-detectable mode and there is a transition of modes from non-detectable to detectable of the monitored hybrid systems. An electrical hybrid system is studied using proposed method with simulation as well as experiment.

The last achievement is the application of QHBG based fault diagnosis framework under single fault assumption in a real world hybrid system - a CyCab mobile robot test-bed. The FDI scheme is applied in the front steering part of the vehicle. The system is very complex because it consists of many electrical, hydraulic and mechanical components. The faults under consideration include parametric faults, sensor faults and fault modes. The hardware structure and software configuration are discussed and the experiment environment is introduced. The FDI decision making scheme is discussed. Three fault scenarios, i.e. an internal leakage fault, a pressure sensor fault and a burnt driver fault, are demonstrated in a real time manner. Experiment results exhibit practical significance of the proposed method.

7.2 Further Research

Based on the works in this thesis, some further research topics are recommended as follows:

In the fault diagnosis method in the thesis, the FDI decision making is static. In other words, once an inconsistency between the model and real system is detected, a set of suspected faults will be created. However, in some situations, different incipient faults have various evolution speeds and their influence on the corresponding residuals will be different. For example, if the system has two incipient faults, one evolves much faster than another one, the faster one might make corresponding residual exceed the predefined residual earlier than the slower one. Therefore, it is not reliable to derive the suspected faults set only based on the first observed inconsistency. It is desirable to set a waiting time for the FDI decision making for reliable fault isolation. Another example is a hybrid with different incipient faults which starts at different modes. Some faults happen at a detectable mode and others at a non-detectable mode. Under this condition, the waiting time should be longer than the time between two mode changes from the time the first observed inconsistency is detected.

The method for the prognosis of multiple failures is realized in an offline manner. For some system, if the component degrades very fast, it is desirable that the prognosis method can be carried out in real time. Therefore, the RUL of the faulty components can be predicted in time to avoid catastrophic effects.

For some hybrid systems, the degradation process will depend on the system operating mode. For example, for some mode, the system will have heavier load than the load in other modes. The faulty components then will deteriorate faster in heavier load. Therefore, the faulty components will have different RUL(s) under different modes. For example, shorter RUL in heavier load and longer one in lighter load.

For the method of prognosis of multiple incipient faults for hybrid systems, the degradation model for each incipient fault is fixed, although unknown, in advance. For some systems, for example, during the lifespan of a bearing, an incipient failure may manifest itself in different stages each one exhibiting different symptoms, from corrosion to spalling, then to cracking. Each symptom can be described by different

degradation models. It is desirable to describe the degradation behavior using multiple degradation models. In this way, the prognosis result can accurately predict the RUL for faulty components under different working condition.

Author's Publications

1. M. Yu, M. Luo, A. Shai, D. Wang, and X. Zhang, "Simultaneous fault and mode switching identification for hybrid systems based on particle swarm optimization," *Expert Systems with Applications*. vol. 37, no. 4, pp: 3000-3012, 2010.
2. M. Yu, D. Wang, M. Luo, and L. Huang, "Prognosis of hybrid systems with multiple incipient faults: augmented global analytical redundancy relations approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* . vol. 41, no. 3, pp: 540-551, 2011.
3. M. Yu, D. Wang, M. Luo, D. Zhang, and Q. Chen, "Fault detection, isolation and identification for hybrid systems with unknown mode changes and fault patterns," *Expert Systems with Applications*, vol. 39, No. 11, pp: 9955-9965, 2012.
4. M. Yu, D. Wang, and S. Arogeti, "GA-based fault parameter identification for hybrid System with unknown mode changes," *The 2nd Int. Symp. Syst. Control in Aeronautics and Astronautics*, pp. 1-5, Dec., 2008, Shenzhen, China.
5. M. Yu, D. Wang, M. Luo, and L. Huang, "Incipient fault diagnosis and prognosis for hybrid systems with unknown mode changes," *IEEE Prognostics & System Health Management Conference*, 2010, Macau.
6. M. Yu, D. Wang, M. Luo, and D. Zhang, "FDI and fault estimation based on differential evolution and analytical redundancy relations," *The 11th International Conference on Control, Automation, Robotics and Vision*, 2010, Singapore.

7. M. Yu, D. Wang, M. Luo, and D. Zhang, "Multiple faults isolation for hybrid systems with unknown fault pattern," *Annual Conference of the PHM Society*, 2010, Portland, USA.
8. M. Yu, M. Luo, S. Arogeti, D. Wang, and X. Zhang, "Fault and mode switching identification for hybrid systems with application to electro-hydraulic system in vehicles", Book Chapter (invited), *Intelligent Unmanned Systems: Theory and Applications. Studies in Computational Intelligence (SCI) series*, Springer-Verlag, 2009.

Bibliography

- [1] S. Simani, C. Fantuzzi, and R. J. Patton. *Model-Based Fault Dagnosis in Dynamic Systems using Identification Techniques*. Springer, 1st edition, 2003.
- [2] C. Ghiaus. Fault diagnosis of air conditioning systems based on qualitative bond graph. *Energy and Buildings*, 30(3): 221-232, August 1999.
- [3] B. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3): 289-338, September 1986.
- [4] B. Kuipers. Qualitative simulation as causal explanation. *IEEE Transactions on Systems Man and Cybernetics*, 17(3): 432-445, 1987.
- [5] T. Gong, and A. L. Tuson. Differential evolution of binary encoding. In *The 11th Online World Conference on Soft Computing in Industrial Applications*, pages 251-262, 18 Sept - 6 Oct 2006.
- [6] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics, Modeling and Simulation of Mechatronics Systems*. New Jersey: John Wiley & Sons Inc., 2006.
- [7] H. T. Ng. Model-based multiple fault diagnosis of time-varying, continuous physical devices. In *the Sixth Conference on Artificial Intelligence Applications*, volume 1, pages 9-15, May 1990.
- [8] S. Subramanian and R. J. Mooney. Qualitative multiple-fault diagnosis of continuous dynamic systems using behavioral modes. In *the 13th National Conference on Artificial Intelligence*, pages 965-970, August 1996.
- [9] R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.
- [10] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. New Jersey: John Wiley & Sons Inc., 2006.

- [11] Q. Shen and R. R. Leitch. Qualitative model based of continuous dynamic systems. In *Proceedings of the First International Conference on Intelligent Systems Engineering*, pages 147-152, Heriot-Watt University, Edinburgh, 1992.
- [12] S. Xia, D.A. Linkens, and S. Bennette. Automatic modeling and analysis of dynamical physical systems using qualitative reasoning and bond graphs. *Intelligent Systems Engineering*, 2(4): 201-212, 1993.
- [13] D. A. Linkens and H. Wang. Fault diagnosis based on a qualitative bond graph model, with emphasis on fault localization. In *IEE Colloquium on Qualitative and Quantitative Modeling Methods for Fault Diagnosis*, pages 1-6, 1995.
- [14] C. H. Lo, Y. K. Wong, and A. B. Rad. Intelligent system for process supervision and fault diagnosis in dynamic physical systems. *IEEE Transactions on Industrial Electronics*, 53(2): 581-592, 2006.
- [15] C. H. Lo, Y. K. Wong, and A. B. Rad. Model-based fault diagnosis in continuous dynamic systems. *ISA Transactions*, 43(3): 459-475, 2004.
- [16] C. H. Lo, Y. K. Wong, and A. B. Rad. Fusion of qualitative bond graph and genetic algorithms: A fault diagnosis application, *ISA Transactions*, 41(4): 445-456, 2002.
- [17] S. X. Ding. *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.
- [18] P. M. Frank and S. X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6): 403-424, 1997.
- [19] R. J. Patton, P. M. Frank, and R. N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 1st edition, 2000.
- [20] E. Isermann. Process fault detection based on modeling and estimation methods-a survey. *Automatica*, 20(4): 387-404, 1984.
- [21] J. Chen and R. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [22] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5): 687-699, 2001.

- [23] J. Chen, R. Pattern, and H. Zhang. Design of unknown input observers and robust fault detection filters. *International Journal of Control*, 63(1): 85-105, 1996.
- [24] S. F. Lin and A. P. Wang. Design of observers with unknown inputs using eignstructure assignment. *International Journal of System Science*, 31(6): 705-711, 2000.
- [25] D. T. Horak. Failure detection in dynamic systems with modelling errors. *Journal of Guidance, Conrol, and Dynamics*, 131(6): 508-516, 1998.
- [26] Z. Shi, F. Gu, B. Lennox, and A. D. Ball. The development of an adaptive threshold for model-based fault detection of a nonlinear electro-hydraulic system. *Control Engineerign Practice*, 13(11): 1357-1367, 2005.
- [27] M. A. Djeziri, R. Merzouki, B. Ould Bouamama, and G. Dauphin-Tanguy. Robust fault diagnosis by using bond graph approach. *IEEE/ASME Tranctions on Mechatronics*, 12(6): 599-611, 2007.
- [28] A. Balluchi, L. Benvenuti, M. D. Di-Benedetto, and A. L. Sangiovanni-Vincetelli. Design of observers for hybrid systems. *Hybrid Systems: Computation and Control, LNCS*, 2289(7): 76-89, 2002.
- [29] W. Wang, L. Li, D. Zhou, and K. Liu. Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 1(6): 2-15, 2007.
- [30] Z. G. Li, C. Y. Wen, and Y. C. Soh. Observer-based stabilization of switching linear systems. *Automatica*, 39(3): 517-524, 2003.
- [31] F. Zhao, X. D. Koutsoukos, H. W. Haussecker, J. Reich, and P. Cheung. Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 35(6): 1225-1240, 2005.
- [32] X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich, and P. Cheung, Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 793-801, 2001.
- [33] F. Zidani, D. Diallo, M. E. H. Benbouzid, and R. Nait-Said. A fuzzy-based approach for the diagnosis of fault modes in a voltage-fed PWM inverter induction motor drive. *IEEE Transactions on Industrial Electronics*, 55(2): 586-593, 2008.

- [34] S. Narasimhan. *Model-Based Diagnosis of Hybrid Systems*. PhD Thesis, Vanderbilt University, 2002.
- [35] S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on System, Man and Cybernetics – Part A: Systems and Humans*, 37(3): 348-361, 2007.
- [36] V. Cocquempot, T. E. Mezyani, and M. Staroswiecki. Fault detection and isolation for hybrid systems using structured parity residuals. In *5th Asian Control Conference*, volume 2, pages 1204-1212, 2004.
- [37] V. Cocquempot, T. E. Mezyani, and M. Staroswiecki. Hybrid dynamical systems monitoring using structured analytical redundancy relations. In *17th IMACS World Congress*, pages 177-183, 2005.
- [38] S. Arogeti, D. Wang, and C. B. Low. Mode identification of hybrid systems in presence of fault. *IEEE Transactions on Industrial Electronics*, 57(4): 1452-1467, 2010.
- [39] C. B. Low, D. Wang, S. Arogeti, and M. Luo. Quantitative hybrid bond graph-based fault detection and isolation. *IEEE Transactions on Automation Science and Engineering*, 7(3): 558-569, 2010.
- [40] C. B. Low, D. Wang, S. Arogeti, and J. B. Zhang. Causality assignment and model approximation for hybrid bond graph: fault diagnosis perspectives. *IEEE Transactions on Automation Science and Engineering*, 7(3): 570-580, 2010.
- [41] C. B. Low, D. Wang, S. Arogeti and M. Luo. Fault parameter estimation for hybrid systems using hybrid bond graph. In *the 3rd IEEE Multi-conference on Systems and Control (MSC 2009)*, pages 1338-1343, 2009.
- [42] S. Arogeti, D. Wang, C. B. Low, D. H. Zhang, and J. Zhou. Mode tracking of hybrid systems in FDI framework. In *the 3rd International Conference on Robotics Automation and Mechatronics (RAM2008)*, pages 841-846, 2008.
- [43] S. Arogeti, D. Wang, and C. B. Low. Mode tracking and FDI of hybrid systems. In *the 10th International Conference on Control, Automation*, pages 112-118, 2008.
- [44] A. Heng, S. Zhang, A. C. C. Tan, and J. Mathew. Rotating machinery prognostics: state of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3): 724-739, 2009.

- [45] H. C. Pusey and M. J. Roemer. An assessment of turbo machinery condition monitoring and failure prognosis technology. *The Shock and Vibration Digest*, 31(2): 365-371, 1999.
- [46] A. K. S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7): 1483-1510, 2006.
- [47] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Wile, Hoboken, NJ, 2006.
- [48] L. Ma. Condition monitoring in engineering asset management, In *Proceedings of Asia-Pacific Vibration Conference*, pages 124-131, Sapporo, Hokkaido, Japan, 2007.
- [49] V. Venkatasubramanian. Prognostic and diagnostic monitoring of complex systems for product lifecycle management: challenges and opportunities. *Computers & Chemical Engineering*, 29(6): 1253-1263, 2005.
- [50] M. Luo, D. Wang, and M. Pham. Model-based fault diagnosis/prognosis for wheeled mobile robots: a review. In *The 31st Annual Conference of IEEE*, pages 2267-2272, 2005.
- [51] Y. Li, T. R. Kurfess, and S. Y. Liang. Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, 14(5): 747-762, 2000.
- [52] A. Ray and S. Tangirala. Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, 4(4): 443-451, 1996.
- [53] D. E. Adams and M. Nataraju. A nonlinear dynamical systems framework for structural diagnosis and prognosis. *International Journal of Engineering Science*, 40(17): 1919-1941, 2002.
- [54] D. Chelidze, J. P. Cusumano, and A. Chatterjee. Dynamical systems approach to damage evolution tracking—Part I: The experimental method. *Journal of vibration and acoustics-Transactions of the ASME*, 124(2): 250-257, 2002.
- [55] J. Qiu, B. B. Set, S. Y. Liang, and C. Zhang. Damage mechanics approach for bearing lifetime prognostics. *Mechanical Systems and Signal Processin*, 16(5): 817-829, 2002.

- [56] J. H. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa. Model-based prognostic techniques applied to a suspension system. *IEEE Transactions on System, Man and Cybernetics – Part A: Systems and Humans*, 38(5): 1156-1168, 2008.
- [57] C. Cempel, H. G. Natke, and M. Tabaszewski. A passive diagnostic experiment with ergodic properties. *Mechanical Systems and Signal Processing*, 11(1):107-117, 1997.
- [58] J. Qiu, C. Zhang, B. B. Seth, and S. Y. Liang. Damage mechanics approach for bearing lifetime prognostics. *Mechanical Systems and Signal Processing*, 16(5): 817-829, 2002.
- [59] G. A. Lesieutre, L. Fang, and U. Lee. Hierarchical failure simulation for machinery prognostics, In *Joint Conference of the 51st Meeting of the SMFPT / 12th Biennial Conference on Reliability, Stress Analysis and Failure Prevention (RSAFP Committe of ASME)*, pages 103-110, 1997.
- [60] S. J. Engel, B. J. Gilmartin, K. Bongort, and A. Hess, Prognostics, the real issues involved with predicting life remaining. In *Proceedings of IEEE Aerospace Conference*, volumn 6, pages 457-469, 2000.
- [61] J. D. Kozlowski, C. S. Byington, A. K. Garga, M. J. Watson, and T. A. Hay. Model-based predictive diagnostics for electrochemical energy sources. In *Proceedings of the IEEE Aerospace Conference*, volumn 6, pages 3149-3164, 2001.
- [62] J. H. Aylor, A. Thieme, and B. W. Johnson. A battery state-ofcharge indicator for electric wheelchairs. *IEEE Transactions on Industrial Electronics*, 39(5): 398-409 1992.
- [63] O. Caumont, P. Le Moigne, C. Rombaut, X. Muneret, and P. Lenain. Energy gauge for lead-acid batteries in electric vehicles. *IEEE Transactions on Energy Conversion*, 15(3): 354-360, 2000.
- [64] B. Zhang, C. Sconyers, R. Patrick, and G. Vachtsevanos. A multi-fault modeling approach for fault diagnosis and failure prognosis of engineering systems. In *Annual Conference of the Prognostics and Health Management Society*, 2009.
- [65] H. M. Paynter. Hydraulics by analog – an electronic model of a pumping plant. *Journal of Boston Society of Civil Engineering*, 46(6): 197-219, 1959.

- [66] D. C. Karnopp, D. Margolis, and R. C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley, 2006.
- [67] P. J. Mosterman and G. Biswas. Behavior generation using model switching: A hybrid bond graph modeling technique. *Transactions of Society for Computer Simulation*, 27(1): 177-182, 1995.
- [68] P. J. Mosterman and G. Biswas. Behaviour generation using model switching-a hybrid bond graph modelling technique. In *International Conference on Bond Graph Modelling*, pages 177-182, 1995.
- [69] P. J. Mosterman and G. Biswas. A theory of discontinuities in physical systems model. *Journal of Franklin Institute*, 335 (3): 401-439, 1998.
- [70] G. M. Asher. The robust modelling of variable topology circuits using bond graphs. In *Proceedings of International Conference on Bond Graph Modelling*, pages 126-131, 1993.
- [71] W. Borutzky, G. Dauphin-Tanguy, and J. U. Thoma. Advances in bond graph modelling: theory, software and applications. *Mathematics and Computers in Simulation*, 39(5-6): 465-475, 1995.
- [72] J. E. Stromberg, J. Top, and U. Soderman. Variable causality in bond graph caused by discrete effects. In *Proceedings of International Conference on Bond Graph Modelling*, pages 115-119, 1993.
- [73] A. C. Umarikar and L. Umanand. Modelling of switched mode power converters using bond graph. *IEEE - Proceedings Electric Power Applications*, 152(1): 51-60, 2005.
- [74] A. C. Umarikar and L. Umanand. Modelling of switching systems in bond graphs using the concept of switched power junctions. *Journal of the Franklin Institute-Engineering and Applied Mathematics*, 342(2): 131-147, 2005.
- [75] A. K. Samantaray and B. Ould Bouamama. *Model-Based Process Supervision: A Bond Graph Approach*. London, U.K.: Springer, 2008.
- [76] I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P. J. Mosterman. A method for efficient simulation of hybrid bond graph. In *Proceeding of Int. Conf. Bond Graph Modeling and Simulation*, volumn 335, pages 177-184, 2007.

- [77] I. Roychoudhury, M. Daigle, G. Biswas, and X. Koutsoukos. Efficient simulation of hybrid systems: an application to electrical power distribution systems. In *Proceeding of the 22nd Eur. Conf. Modeling Simulation*, pages 471-477, 2008.
- [78] M. Tagina, J. P. Cassar, G. Dauphin-Tanguy, and M. Staroswiecki. Monitoring of systems modelled by bond-graphs. In *Proceeding of Int. Conf. Bond Graph Modeling*, pages 275-279, 1995.
- [79] B. Ould Bouamama, A. K. Samantaray, M. Staroswiecki, and G. Dauphin-Tanguy. Derivation of constraint relations from bond graph models for fault detection and isolation. In *Proceeding of Int. Conf. Bond Graph Modeling*, pages 104-109, 2003.
- [80] A. K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Component-based modelling of thermofuild systems for sensor placement and fault detection. *Simulation*, 80(2): 381-398, 2004.
- [81] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998.
- [82] B. Ould Bouamama, K. Medjaher, M. Bayart, A. K. Samantaray, and B. Conrard. Fault detection and isolation of smart actuators using bond graphs and external model. *Control Engineering Practice*, 13(2): 159-175, 2005.
- [83] A. S. Wilsky, E. Y. Chow, X. C. Lou, and G. C. Verghese. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7): 603-614, 1984.
- [84] M. Staroswiecki. Quantitative and qualitative models for fault detection and isolation. *Mechanical Systems and Signal Processing*, 14(3): 301-325, 2000.
- [85] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5): 687-699, 2001.
- [86] I. Hwang, S. Kim, Y. Kim, C. E. Seah. A Survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*. 18(3): 636-653, 2010.
- [87] S. Anwar and L. Chen. An analytical redundancy-based fault detection and isolation algorithm for a road-wheel control subsystem in a steer-by-wire eystem. *IEEE Transactions on Vehicular Technology*, 56(5): 2859-2868, 2007.

- [88] M. A. Djeziri, R. Merzouki, and B. Ould Bouamama. Robust monitoring of an electric vehicle with structured and unstructured uncertainties. *IEEE Transactions on Vehicular Technology*, 58(9): 4710-4719, 2009.
- [89] C. H. Lo, Y. K. Wong, and A. B. Rad. Bond graph based Bayesian network for fault diagnosis. *Applied Soft Computing*, 11(1): 1208-1212, 2011.
- [90] A. Moustafa, S. Mahadevan, and M. Daigle. Structural and sensor damage identification using the bond graph approach. *Structural Control & Health Monitoring*, 17(2): 178-197, 2010.
- [91] K. Fawaz, R. Merzouki, and B. Ould Bouamama. Model based real time monitoring for collision detection of an industrial robot. *Mechatronics*, 19(5): 695-704, 2009.
- [92] M. A. Djeziri, B. Ould Bouamama, and R. Merzouki. Modelling and robust FDI of steam generator using uncertain bond graph model. *Journal of Process Control*, 19(1): 149-162, 2009.
- [93] R. Merzouki, K. Medjaher, M. A. Djeziri, and B. Ould Bouamama. Backlash fault detection in mechatronic system. *Mechatronics*, 17(6): 299-310, 2007.
- [94] M. J. Daigle, X. D. Koutsoukos, and G. Biswas. Distributed diagnosis in formations of mobile robots. *IEEE Transactions on Robotics*, 23(2): 353-369, 2007.
- [95] M. Khemliche, B. Ould Bouamama, and H. Haffaf. Sensor placement for component diagnosability using bond-graph. *Sensors and Actuators A-Physical*, 132(2): 547-556, 2006.
- [96] C. B. Low, D. Wang, S. Arogeti, and J. B. Zhang. Monitoring ability and quantitative fault diagnosis using hybrid bond graph. In *the 17th IFAC World Congress*, pages 10516-10521, 2008.
- [97] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE Proceeding of Int. Conf. on Neural Networks*, volume 4, pages 1942-1948, 1995.
- [98] J. Kennedy and R. Eberhart. A discrete binary version of the particle swarm optimisation algorithm. In *IEEE Proceeding of Int. Conf. on Neural Networks*, volume 4, pages 4104-4108, 1997.
- [99] D. E. Goldberg. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Boston: Addison-Wesley, 1989.

- [100] M. Gen and R. Cheng. *Genetic Algorithm & Engineering Optimization*. New York: John Wiley & Sons, 2000.
- [101] Y. Shi, and R. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1945-1950, 1999.
- [102] A. Chatterjee, and P. Siarry. Nonlinear inertial weight variation for dynamic adaption in particle swarm optimization. *Computer and Operations Research*, 33(3): 859-871, 2004.
- [103] G. Chen, X. Huang, J. Jia, and Z. Min. Natural exponential inertia weight strategy in particle swarm optimization. In *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pages 3672-3675, 2006.
- [104] H. E. Merritt. *Hydraulic Control Systems*. New York: Wiley, 1967.
- [105] H. Chen, P. Y. Sun, and K. H. Guo. A multi-objective control design for active suspensions with hard constraints. In *Proceedings of the American Control Conference*, volumn 5, pages 4371-4376, 2003.
- [106] H. Chen, and K. H. Guo. Constrained H_∞ control of active suspensions: an LMI approach. *IEEE Transactions on Control Systems Technology*, 13(3): 412-421, 2005.
- [107] L. K. Mak, Y. S. Wong, and X. X. Wang. An adaptive genetic algorithm for manufacturing cell formation. *International Journal of Advance Manufacture Technology*, 16(7): 491-497, 2000.
- [108] J. Wang, J. D. Wang, N. Q. Daw, and H. Wu. Identification of pneumatic cylinder friction parameters using genetic algorithms. *IEEE/ASME Transactions on Mechatronics*, 9(1): 100-107, 2004.
- [109] M. Yu, D. Wang, Ming Luo, and Danhong Zhang. Multiple faults isolation for hybrid systems with unknown fault pattern. In *Proceedings of Annual Conference of the Prognostics and Health Management Society*. Portland, USA, 2010.
- [110] R. Storn and K. Price. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4): 341-359, 1997.

- [111]R. Storn, and K. Price. Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report TR-95-012*, International Computer Science Institute, Berkley, 1995.
- [112]D. G. Kurup, M. Himidi, and A. Rydberg. Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm. *IEEE Transactions on Antennas Propagation*, 51(9): 2210-2217, 2003.
- [113]Z. Fan, J. Liu, T. Sorensen, and P. Wang. Improved differential evolution based on stochastic ranking for robust layout synthesis of MEMS components. *IEEE Transactions on Industrial Electronics*, 56(4): 937-948, 2009.
- [114]L. S. Coelho and V. C. Mariani. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power Systems*, 21(2): 989-996, 2006.
- [115]K. Price. Genetic annealing. *Dr. Dobbs's Journal*, 220(2): 127-132, 1994.
- [116]V. Feoktistov. *Differential Evolution: In Search of Solutions*. Springer, USA, 2006
- [117]E. Mezura-Montes, J. Velazquez-Reyes, and C. A. C. Coello. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 2006 conference on Genetic and evolutionary computation*, pages 485-492, 2006.
- [118]W. Batko. *Prediction Method in Technical Diagnostics*. PhD Thesis, Cracov Mining Academy, 1984.
- [119]K. Kazmierczak. Application of autoregressive prognostic techniques in diagnostics, In *Proceedings of the Vehicle Diagnostics Conference*, pages 128-134, 1983.
- [120]W. Wang, F. Golnaraghi, and F. Ismail. Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, 18(4): 813-831, 2004.
- [121]W. Wang. An adaptive predictor for dynamic system forecasting. *Mechanical Systems and Signal Processing*, 21(2): 809-823, 2007.
- [122]N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran. Residual life predictions from vibration-based degradation signals: a neural network approach. *IEEE Transactions on Industrial Electronics*, 51(3): 694-700, 2004.

- [123]N. Gebraeel, and M. Lawley. A neural network degradation model for computing and updating residual life distributions. *IEEE Transactions on Automation Science and Engineering*, 5(1): 154-163, 2008.
- [124]C. J. Li, and S. Choi. Spur gear root fatigue crack prognosis via crack diagnosis and fracture mechanics. In *Proceedings of the 56th Meeting of the Society of Mechanical Failures Prevention Technology*, pages 311-320, Virginia Beach, VA, 2002.
- [125]C. J. Li, and H. Lee. Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics. *Mechanical Systems and Signal Processing*, 19(3): 836-846, 2005.
- [126]R. F. Orsagh, J. Sheldon, and C.J. Klenke. Prognostics/diagnostics for gas turbine engine bearings. In *Proceedings of the IEEE Aerospace Conference*, volume 7, pages 3095-3103, 2003.
- [127]G. J. Kacprzynski, A. Sarlashkar, M.J. Roemer, A. Hess, and B. Hardman. Predicting remaining life by fusing the physics of failure modeling with diagnostics. *JOM Journal of the Minerals, Metals and Materials Society*, 56(2): 29-35, 2004.
- [128]D. E. Adams and M. Nataraju. A nonlinear dynamical systems framework for structural diagnosis and prognosis. *International Journal of Engineering Science*, 40(17): 1919-1941, 2002.
- [129]Z. L. Gaing. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, 19(2): 384-391, 2004.
- [130]S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan. Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging. *IEEE Transactions Industrial Electronics*, 55(9): 3447-3460, 2008.
- [131]A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi. A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems. *IEEE Transactions Industrial Electronics*, 52(6): 1478-1489, 2005.

- [132]Z. L. Gaing. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Transactions on Power Systems*, 18(3): 1187-1195, 2003.
- [133]S. Simania and R. J. Pattonb. Fault diagnosis of an industrial gas turbine prototype using a system identification approach. *Control Engineering Practctice*, 16(7): 769-786, 2008.
- [134]J. Watton. *Modelling, Monitoring and Diagnostic Techniques for Fluid Power Systems*. Springer, London, 2007.
- [135]M. Orchard and G. Vachtsevanos. A particle filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, 31(3): 221-246, 2009.
- [136]P. Wang, and G. Vachtsevanos. Fault prognostics using dynamic wavelet neural networks. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(4): 349-365, 2001
- [137]J. W. Sheppard, and M. A. Kaufman. Bayesian diagnosis and prognosis using instrument uncertainty. In *Proceedings of AUTOTESTCON*, pages 417-423, 2005.
- [138]C. Bunks, D. McCarthy, and T. Al-Ani. Condition-based maintenance of machines using hidden Markov models. *Mechanical Systems and Signal Processing*, 14(4): 597-612, 2000.
- [139]P. Baruah, and R. B. Chinnam. HMMs for diagnostics and prognostics in machining processes. *International Journal of Production Research*, 43(6):1275-1293, 2005.
- [140]X. D. Zhang, R. Xu, K. Chiman, S. Y. Liang, Q. L. Xie, and L. Haynes. An integrated approach to bearing fault diagnostics and prognostics. In *Proceedings of the American Control Conference*, volume 4, pages 2750-2755, 2005.
- [141]C. Kwan, X. Zhang, R. Xu, and L. Haynes. A novel approach to fault diagnostics and prognostics. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volumn 1, pages 604-609, 2003.
- [142]H. Ding, X. Gui, and S. Yang, An approach to state recognition and knowledge-based diagnosis for engines. *Mechanical Systems and Signal Processing*, 5(4): 257-266, 1991.

- [143] X. Lou, and K. A. Loparo. Bearing fault diagnosis based on wavelet transform and fuzzy inference. *Mechanical Systems and Signal Processing*, 18(5): 1077-1095, 2004.
- [144] M. C. Pan, P. Sas, and H. Van Brussel, Machine condition monitoring using signal classification techniques. *Journal of Vibration and Control*, 9(10): 1103-1120, 2003.
- [145] B. M. Ebrahimi, and J. Faiz. Feature extraction for short-circuit fault detection in permanent-magnet synchronous motors using stator-current monitoring. *IEEE Transactions on Power Electronic*, 25(10): 2673-2682, 2010.
- [146] Q. Wu. Fault diagnosis model based on Gaussian support vector classifier machine. *Expert Systems with Applications*, 37(9): 6251-6256, 2010.
- [147] J. I. Park, S. H. Baek, M. K. Jeong, and S. J. Bae. Dual features functional support vector machines for fault detection of rechargeable batteries. *IEEE Transactions on Systems, Man and Cybernetics Part C-Applications and Reviews*, 39(4): 480-485, 2009.
- [148] Y. Yao, and F. R. Gao. Statistical monitoring and fault diagnosis of batch processes using two-dimensional dynamic information. *Industrial & Engineering Chemistry*, 49 (20): 9961-9969, 2010.
- [149] Q. P. He, and J. Wang. Large-scale semiconductor process fault detection using a fast pattern recognition-based method. *IEEE Transactions on Semiconductor Manufacturing*, 23(2): 194-200, 2010.
- [150] B. Li, M. Y. Chow, Y. Tipsuwan, and J.C. Hung, Neural-network-based motor rolling bearing fault diagnosis, *IEEE Transactions on Industrial Electronic*, 47(5): 1060-1069, 2000.
- [151] Y. Fan, and C. J. Li. Diagnostic rule extraction from trained feedforward neural networks. *Mechanical Systems and Signal Processing*, 16(6): 1073-1081, 2002.
- [152] J. K. Sporre. Application of the cascade correlation algorithm (CCA) to bearing fault classification problems. *Computers in Industry*, 32(3): 295-304, 1997.
- [153] C. C. Wang, and G. P. J. Too. Rotating machine fault detection based on HOS and artificial neural networks. *Journal of Intelligent Manufacturing*, 13(4): 283-293, 2002.

- [154]R. M. Tallam, T. G. Habetler, and R. G. Harley. Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics. *IEEE Transactions on Power Electronics*, 17(6): 1089-1095, 2002.
- [155]M. Stanek, M. Morari, and K. Frohlich. Model-aided diagnosis: An inexpensive combination of model-based and case-based condition assessment. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 31(2): 137-145, 2001.
- [156]B. S. Yang, T. Han, and Y. S. Kim. Integration of ART-Kohonen neural network and case-based reasoning for intelligent fault diagnosis, *Expert Systems with Applications*, 26(3): 387-395, 2004.
- [157]S. Sampath, S. Ogaji, R. Singh, and D. Probert. Engine-fault diagnostics: an optimisation procedure. *Applied Energy*, 73(1): 47-70, 2002.
- [158]B. Samanta and C. Nataraj. Use of particle swarm optimization for machinery fault detection. *Engineering Applications of Artificial Intelligence*, 22(2): 308-316, 2009.
- [159]Q. Wu. Car assembly line fault diagnosis based on robust wavelet SVC and PSO. *Expert Systems with Applications*, 37(7): 5423-5429, 2010.
- [160]C. Sueur and G. Dauphin-Tanguy. Bond graph approach to multi scale systems analysis. *Journal of the Franklin Institute*, 328: 1005–1026, 1991.
- [161]R. C. Rosenberg, and D. C. Karnopp. A definition of the bond graph language. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 94(3): 179-182, 1972.
- [162]L. A. Zadeh. Fuzzy set. *Information and Control*, 8: 338-353, 1965.
- [163]E. H. Mamdani. Application of fuzzy algorithms for simple dynamic plant. *Proceeding of IEE*, 121: 1585-1588, 1974.
- [164]M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Berlin, Germany: Springer-Verlag, 2003.
- [165]J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5(5): 653-661, 1997.
- [166]R. J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. In *Proceedings of the First IFAC-IMACS Symposium SAFEPROCESS*, volumn 1, pages 239-255, Hull, UK, 1991.