

On-line Fault Diagnosis Using Signed Digraphs

by

Liqiang Wang

B. Eng Tianjin University, China, 1997

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF**

Master of Science in Engineering

In the Graduate Academic Unit of Department of Electrical and Computer
Engineering

Supervisor(s): Dr. J. H. Taylor, PhD
Dept. of Electrical & Computer Engineering
Examining Board: Dr. C. Diduch, PhD
Dept. of Electrical & Computer Engineering
Dr. R. Doraiswami, PhD
Dept. of Electrical & Computer Engineering
External Examiner: Mr. J. Dedourek, Faculty of Computer Science

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

January, 2006

©Liqiang Wang, 2006

To My Loved Family

Abstract

Due to the increasing complexity and necessity of safety and economy in industrial processes, efficient diagnosis systems are becoming more and more important. A wide variety of different approaches have already been proposed in the literature. The signed directed graph (SDG) approach has been a widely studied diagnostic strategy, mainly because of its resemblance to a human way of reasoning, its in-depth analysis of abnormal situation, and its ability to present a complete picture of all possible fault hypotheses. However the SDG diagnosis approach focuses on qualitative information of the process, so unavoidable diagnosis restrictions and difficulties to achieve resolution exist. In this thesis, an intelligent fault diagnosis algorithm based on an SDG qualitative model approach is presented. The algorithm combines quantitative process information into an SDG propagation model and uses additional process knowledge to improve performance. Both initial and ultimate response fault patterns are used to isolate the root fault, improving diagnosis resolution, and facilitating the early detection and diagnosis of process faults.

Acknowledgements

First of all and foremost, I would like to thank and express my gratitude to Dr. James Taylor, my supervisor, for his advice, encouragement, many inspiring suggestions and constant support throughout my graduate studies.

I would like to thank Atalla Sayda, our team leader, for the support, advice and help in many ways during the research. I would also like to thank Mano Ram Maurya, from San Diego Supercomputer Center, a never met person, for his kind help and nice suggestions when I began this totally new area. I would also like to thank my friends, James Wang, Nan Feng, Lydia Hua, and many others, for their help in many ways. I would like to express my sincerest gratitude towards them.

I am grateful to my loving husband and my lovely daughter, for the happiness and brightness they brought to me throughout the hard days. I would express my gratitude to my parents, for their help in my life and encouragement for my study in Canada. Without their love and sacrifice, I could not have finished my study abroad.

Table of Contents

Abstract	iii
Acknowledgments	iv
Table of Contents	v
List of Tables	x
List of Figures	xiv
Nomenclature	xv
1 Introduction	1
1.1 Motivation and Scope	1
1.2 Fault Detection and Diagnosis	2
1.2.1 Brief Review of Diagnosis Techniques	3
1.2.2 Desired Attributes of a Fault Diagnosis System	4
1.3 Objective and Thesis Organization	6
1.3.1 Objective	6

1.3.2	Thesis Organization	7
2	Literature Review	8
2.1	Introduction to SDG	8
2.2	SDG Approach Development	9
2.3	Attempts to Add Quantitative Information	11
2.4	Recent Activity	12
3	Development of the Diagnostic System	14
3.1	Introduction to SDG-based Fault Diagnosis	16
3.2	Building the SDG Model of the System	19
3.3	Form of the Diagnosis Knowledge Base	22
3.4	Deriving Patterns from an SDG	26
3.4.1	Initial Response	27
3.4.2	Ultimate Response	29
3.5	Combining Quantitative Data	30
3.6	Online Fault Detection and Diagnosis	31
3.7	Fault Detection	32
3.8	Conclusion	34
4	Implementation and Simulation Results	35
4.1	System Model	35
4.1.1	JCSTR Dynamic Model	35

4.1.2	JCSTR SDG Model	37
4.2	Fault Analysis	39
4.2.1	Disturbance Caused by Mix Inflow Rate F_i	41
4.2.1.1	Initial Response Fault Patterns for $F_i(+)$	42
4.2.1.2	Ultimate Response Fault Patterns for $F_i(+)$	44
4.2.1.3	Fault Patterns for $F_i(-)$	45
4.2.2	Fault Caused by Mix Inlet Temperature T_i	45
4.2.2.1	Fault Patterns for High Inlet Temperature $T_i(+)$	46
4.2.2.2	Fault Patterns for Low Inlet Temperature $T_i(-)$	47
4.2.3	Fault Caused by Heating Inlet Temperature T_{ji}	47
4.2.3.1	Fault Patterns for High Heating Inlet Temperature $T_{ji}(+)$	47
4.2.3.2	Fault Patterns for Low Heating Fluid Temperature $T_{ji}(-)$	49
4.2.4	Volume Sensor Fault, V_s bias	49
4.2.4.1	Initial Response Fault Pattern for $V_s(-)$	50
4.2.4.2	Ultimate Response Fault Pattern for $V_s(-)$	50
4.2.5	Mix Outflow Valve Fault, O_v Stuck	51
4.2.5.1	Initial Response Fault Pattern for $O_v(-)$	51
4.2.5.2	Ultimate Response Fault Pattern for $O_v(-)$	52
4.2.6	Temperature Sensor Fault, T_s Bias	52
4.2.6.1	Initial Response Fault Pattern for $T_s(-)$	52

4.2.6.2	Ultimate Response Fault Pattern for $T_s(-)$	53
4.2.7	Heating Inflow Valve Fault, I_v stuck	53
4.2.7.1	Initial Response Fault Pattern for $I_v(-)$	53
4.2.7.2	Ultimate Response Fault Pattern for $I_v(-)$	54
4.2.8	Summary of Fault Patterns	54
4.3	Simulation Results - Purely Qualitative FDI	55
4.4	Hybrid FDI Simulation Results	59
4.5	Conclusion	61
5	Robustness of the Fault Diagnostic System	62
5.1	Different Thresholds	63
5.2	Dealing with Transients	65
5.3	Operation Point Variation	68
5.4	Operation Fluctuation	69
5.5	Conclusion	72
6	Conclusions and Suggestion for Future Developments	74
6.1	Main Conclusions	74
6.2	Suggestions for Future Work	76
	Bibliography	78
	A SDG Concepts	81
	B Necessary Conditions for CVs and IVs	84

C	Simulation Results for Robustness Tests	85
C.1	Simulation Results for Transient Operation	85
C.2	Simulation Results for Operation Point Variations	87
C.3	Simulation Result for Operation Fluctuation	89
D	Matlab Code for Simulation	98
Vita		112

List of Tables

4.1	Parameters and Variables of JCSTR Model	37
4.2	Arc Value Calculation for DE Equations	39
4.3	Fault Origin List	40
4.4	Measurements for On-line Diagnosis	40
4.5	Possible Patterns Table for JCSTR Model	55
5.1	Comparison Two Sets of Thresholds	63
5.2	Four Cases of Operation Fluctuation	70
5.3	FDI Results for Robustness Test, 1	71
5.4	FDI Results for Robustness Test, 2	72
5.5	Thresholds Used in Robustness Tests	72

List of Figures

3-1	Architecture for the Fault Diagnostic System	15
3-2	Example of SDG Model	17
3-3	SDG for a Control Loop	22
3-4	Implementation of the Algorithm	26
4-1	JCSTR Model and Controller	36
4-2	SDG Model for JCSTR Model	38
4-3	Reduced SDG Model for Mix Inflow Fault	41
4-4	SDG Model with Arc Lengths	42
4-5	T_j Response when F_i High, $t_{ofault} = 1.5$	43
4-6	SDG Model for Fault T_i	45
4-7	T_j Response when T_i High, $t_{ofault} = 1.5$	46
4-8	SDG Model for Fault T_{ji}	47
4-9	T_j Response when T_{ji} High, $t_{ofault} = 1.5$	48
4-10	SDG Model for V_s Fault	49
4-11	SDG Model for O_v Fault	51

4-12	SDG Model for T_s Fault	52
4-13	SDG Model for I_v Fault	54
4-14	FDI Results for Fault F_i	56
4-15	FDI Results for Fault T_i	57
4-16	FDI Results for Fault T_{ji}	57
4-17	FDI Results for Sensor Faults	58
4-18	FDI Results for Valve Faults	58
4-19	Different Behavior of V for Fault F_i low and $V_s(-)$	60
4-20	FDI Results with Hybrid FDD	60
4-21	FDI Results with Hybrid FDD	61
5-1	FDI Result, Different Thresholds for Fault Size 10%	64
5-2	FDI Result, Different Thresholds for Fault Size 20%	64
5-3	FDI Result, Different Thresholds for Normal Operation	66
5-4	FDI Result, Fault Happened in Transient Period	67
5-5	FDI Work Region	68
5-6	Example FDI Results	69
5-7	Example FDI Results	71
A-1	Truth Table for Qualitative Simulation	81
C-1	FDI Results for Fault Happening in a Transient	85
C-2	FDI Results for Fault Happening in a Transient	86

C-3 FDI Results for Fault Happening in a Transient	86
C-4 FDI Results for Fault Happening in a Transient	86
C-5 FDI Results for Fault Happening in a Transient	87
C-6 FDI Work Region	87
C-7 FDI Work Region	88
C-8 FDI Work Region	88
C-9 FDI Work Region	88
C-10 FDI Results for System with Fluctuation, Case A	89
C-11 FDI Results for System with Fluctuation, Case A	89
C-12 FDI Results for System with Fluctuation, Case A	90
C-13 FDI Results for System with Fluctuation, Case A	90
C-14 FDI Results for System with Fluctuation, Case A	90
C-15 FDI Results for System with Fluctuation, Case B	91
C-16 FDI Results for System with Fluctuation, Case B	91
C-17 FDI Results for System with Fluctuation, Case B	92
C-18 FDI Results for System with Fluctuation, Case B	92
C-19 FDI Results for System with Fluctuation, Case B	92
C-20 FDI Results for System with Fluctuation, Case C	93
C-21 FDI Results for System with Fluctuation, Case C	93
C-22 FDI Results for System with Fluctuation, Case C	94
C-23 FDI Results for System with Fluctuation, Case C	94

C-24 FDI Results for System with Fluctuation, Case C	95
C-25 FDI Results for System with Fluctuation, Case D	95
C-26 FDI Results for System with Fluctuation, Case D	96
C-27 FDI Results for System with Fluctuation, Case D	96
C-28 FDI Results for System with Fluctuation, Case D	97
C-29 FDI Results for System with Fluctuation, Case D	97

Nomenclature

A,B,C,D	node
A	area for heat transfer, magnitude
c_p	heat capacity
F	volumetric flow rate
I	heating inflow
O	outflow
Q	rate of heat transfer
T	temperature, period
U	heat transfer coefficient
V	volume, valve
t	time
ρ	density

Subscripts

i	inlet
j	jacket
ji	jacket inlet
s	sensor
v	control valve
sp	set point

Abbreviations

AE	algebraic equation
CR	compensatory response
CSTR	continuous stirred tank reactor
CV	compensatory variable
DAE	differential algebraic equation
DCS	distributed control system
DE	differential equation
ESDG	extended signed directed graph
FDD	fault detection and diagnosis
FDI	fault detection and isolation
IR	inverse response
IV	inverse variable
JCSTR	jacketed continuous stirred tank reactor
LC	level control
MSCC	maximum strongly connected component
PC	personal computer
PCA	principle component analysis
QTA	quality trend analysis
SCC	strongly connected component
SDG	signed directed graph
TC	temperature control

Chapter 1

Introduction

1.1 Motivation and Scope

With the advent of computer control, the scope of process control has made significant changes in the last three decades. Regulatory control, the lower level control, such as opening and closing valves, which used to be performed by operators, is now done by automated control with the aid of computers with considerable success, so it is not now a focus of advanced control. Upper level control, intelligent supervisory control, is becoming the new focus of industrial and academic researchers. An intelligent supervisory control system is a system that makes the best use of conventional control approaches and artificial intelligence to optimally manage the plant process, evaluating local controllers' performance, diagnosing causes for abnormal situations, planning actions and executing the planned actions. Typical goals of a supervisory control system are safe operation, highest product quality and most economic operation, thus making fault detection and diagnosis (FDD) a critical control task.

Abnormal situations happen in chemical and industrial processes when processes deviate significantly from their normal operation ranges, usually due to sensor drifts,

equipment failures, changes in process parameters or operator errors; these events are also called faults or failures in the literature. If these abnormal situations are ignored, there are serious consequences and even accidents, such as fires or explosions may occur. Even when no such emergencies occur, they do cause low product quality, equipment damage, and other significant costs.

Responding to abnormal situations in process plant mainly depends on manual activity, and is performed by on site operators. When an abnormal situation occurs, the operator is warned by flashing or buzzing devices, each of which corresponds to one point or element of the plant, and the operator should quickly make a correct judgment about the root cause and take actions responding to the failure events to return the system to its normal operation. However, due to the size and complexity of industrial plants, the variety of faults, and operators' limited knowledge, experience and ability to handle stress, human operators may make erroneous decisions and take actions that make the fault even worse. The complete reliance on human operators is not suitable for modern process plants. Therefore, development of effective computer-aided on-line fault diagnosis techniques in order to keep the system performance as close as possible to the optimal condition has become an important issue for plant operation.

1.2 Fault Detection and Diagnosis

First of all, we must define the terms of fault and fault diagnosis in the context of the process industry. The words fault and malfunction are used in relation to equipment as synonyms to designate the departure of an observed variable or calculated parameter from an acceptable range [1]. Thus a fault may be a process abnormality or symptom, such as high pressure in a reactor or low voltage in a power system. Fault

detection is used to identify whether or not the process (measured) variables, individually or collectively, are within their normal range. Hence, the notion of normal operation region is an important concept in fault detection. Fault diagnosis refers to the determination (after detection of a fault occurrence) of the equipment, or portion of it that is causing the fault(s) [1]. That is, diagnosis is the determination of the fault origin (the root cause of a fault).

In general, faults can be classified in three categories:

- external disturbances from the environment,
- equipment failures (stuck valves, leaking pipes), and
- malfunctioning sensors and actuators.

1.2.1 Brief Review of Diagnosis Techniques

Developing automated fault detection and diagnosis using intelligent control approaches has been studied by researchers, and various computer aided approaches have been developed over the years. These methods can be divided into three classifications: quantitative model based methods (parameter estimation), qualitative model based methods (digraphs, fault trees), and process history based methods (Quality Trend Analysis, QTA; Principle Component Analysis, PCA) [2].

Quantitative model based methods (such as observers, extended Kalman filters) are residual based methods, the residual representing the difference between various functions of the outputs and the expected values of the functions under normal conditions. They heavily depend on accurate mathematical relations between variables, and the models are built from first principles, such as mass, energy and momentum balances, or model identification approaches. The theory for linear models is well developed,

however for a general nonlinear model, the linear approximation is used but the effectiveness of these methods could be reduced.

Qualitative model based methods (such as digraphs, fault trees) use causal models representing the cause-effect relationship between system variables, so system behavior can be predicted by the model. Forward or backward reasoning is used to find all possible fault candidates, similar to the way a human solves problems, and explanation generation is relatively straightforward, making them more interactive. They do not need as extensive information as quantitative model based methods do. However, poor resolution due to the nature of qualitative models has been addressed by many researchers.

Process history based methods (expert systems, neural networks) only need large amounts of historical process data, which can be transformed and presented as *a priori* knowledge to the diagnostic system. This knowledge can be either qualitatively or quantitatively extracted. They can be model based or may not need a model. Since process history approaches are easy to implement and require little modeling and *a priori* knowledge, they are widely applied in process industries. The limitation of process history based methods is in the availability of measurements and the application is mainly restricted to sensor faults.

Venkatasubramanian and coworkers present a more comprehensive review of the various faults diagnosis methods. Interested readers please see references [2, 3, 4].

1.2.2 Desired Attributes of a Fault Diagnosis System

There are different criteria to judge a diagnosis system, but they mainly include isolability, early detection and diagnosis, and robustness. These quality properties of failure isolation system strongly influence the usefulness of such system.

Isolability is the ability of a diagnosis system to distinguish or isolate certain specific faults from each other, given that the fault size is large enough. Isolability is a tradeoff problem between completeness and high resolution. Whenever a fault happens, a diagnosis system produces hypotheses for the abnormal event, the true fault should be a subset of the hypotheses to meet the requirement for completeness, but for high resolution, the subset should be as minimal as possible.

Early detection and diagnosis is an important desirable attribute of a diagnosis system. It is closely related to sensitivity, which characterizes the size of faults that can be isolated under certain conditions. It should be realized that there exists a conflict between quick response to faults and the tolerance of performance variation of the system during normal operating conditions.

Robustness is the most critical requirement by FDD systems. A robust diagnosis system should not fail totally and abruptly when faced with various noise or uncertainties during operation, instead it should degrade gracefully. For example, in ideal operation, the FDD thresholds may set be close to zero, but in the presence of noise, these thresholds must be chosen more conservatively, to avoid robustness problems.

Other important characteristics of a fault diagnosis system also include novelty identifiability, i.e., the ability to recognize the occurrence of novel faults and not misclassify them as one of the other known faults or normal operation; and adaptability, e.g., the diagnostic system should be adaptable to changes in external inputs or structural changes. An explanation facility, to explain how the fault originated and propagated to the current situation is also helpful. Modeling requirements, i.e., the amount of modeling required for the development of a diagnostic system, or real-time diagnostic classifier is a major issue: The modeling effort should be as minimal as possible, as should storage and computational requirements; there is a tradeoff between computational complexity of the diagnostic methodology and storage requirements. Finally,

multiple fault identifiability, the ability to identify multiple faults is an important but difficult requirement.

Usually there is no single diagnostic technique or method that can meet all the above characteristics. Each method has its own advantages and disadvantages depending on its usage, and no one is superior to others in general. Recently, with the growing maturity of many kinds of techniques, hybrid methods, which combine different approaches have attracted some researchers. It is believed that several approaches can complement one another, and a hybrid FDD method can overcome the limitations of individual strategies.

1.3 Objective and Thesis Organization

1.3.1 Objective

Developing an effective and intelligent computer-aided on-line fault diagnosis technique is the objective of this thesis.

Fault detection and isolation (FDI) is very important in achieving high industry product quality, preventing damage to production facilities, and protecting the personal safety of the operators. When a fault appears, it should be detected as early as possible and the fault should be determined correctly as well. The purpose of this thesis is to develop an intelligent algorithm that can fulfill the requirement of early detection and accurate diagnosis of process faults. An intelligent system combining a Signed Digraph (SDG) based approach with process knowledge is proposed. This FDI algorithm will be used in several intelligent control projects currently supervised by Dr. Taylor.

This thesis is focused on the SDG approach, which is a qualitative technique. Two

other team members are using quantitative and data history based methods respectively. The final goal of our research team is to develop a hybrid diagnosis strategy for process industry, by combining each optimized individual approach.

1.3.2 Thesis Organization

In Chapter 2, a brief literature survey of existing research and various methods for SDG based fault diagnosis and analysis is presented. The benefits and drawbacks of the SDG based FDD approach are discussed. Based on a study of the literature, an algorithm is proposed in chapter 3. This algorithm takes advantage of the benefits of SDG, and overcomes its shortcomings, to provide an efficient on-line fault diagnostic approach. Chapter 4 presents simulation tests for the proposed algorithm. The simulation model is a nonlinear Jacketed Continuously Stirred Tank Reactor (JCSTR) model. The robustness of the approach is discussed in chapter 5. The thesis concludes with a discussion of the approach, and gives some suggestions for future work, which composes chapter 6.

Chapter 2

Literature Review

2.1 Introduction to SDG

In the recent past, SDG-based approaches have been proposed by various researchers for fault diagnosis of chemical process systems, and they have shown promise. A Signed Directed Graph or Signed Digraph (SDG), as a qualitative model, effectively and graphically represents a process system. Nodes in a digraph correspond to state variables, failure origins and alarm conditions; and directed arcs between the nodes show the cause-effect relationship between these variables. Each node can take a value of (0), (+), and (-), representing that the corresponding variable is at its normal steady state value, above or below the nominal steady state value, respectively. Arcs take values (+), or (-), indicating that the cause and effect change is in the same direction or the opposite direction. An SDG model shows the pathways of causality for fault propagation.

Unlike quantitative approaches, which require a rigorous process model and extensive measurements to collect process data for parameter estimation, an SDG-based approach requires only a minimum of process data to perform a quick diagnosis. It

is able to deal with uncertainty, incomplete information and noise. Since the process system is expressed causally as a digraph model, an SDG also facilitates reasoning, and fault explanation. However, since SDG approaches only use qualitative information, the low of diagnostic resolution is a common drawback.

2.2 SDG Approach Development

Using signed digraphs (SDGs) for fault diagnosis was first proposed by Iri et al. in 1979 [5]. An attempt was made to apply graph theory to diagnose system failures. The influences of system elements are represented by a signed digraph model and the notion of patterns (composed of the node values) on the model is introduced to represent symptoms of the system under abnormal situation. The origin of the system failure can be located at the maximal strongly connected component in the cause-effect graph. In the following year, Umeda et al. developed a method for cause and effect analysis of a processing system based on SDGs [6]. Variations on the SDG involving multiple time stages and delay time have been addressed. Methods for obtaining an SDG from differential equations and algebraic equations were also presented in the same article. A diagnostic algorithm presented later by Kokowa et al. [7] incorporated delays, gains, and fault propagation probability into the digraph, but the method only applied to processes without feedback. Shiozaki and his coworkers [8] also extended the idea of SDG into five range patterns (0, +, +?, -, and -?) instead of three range pattern (0, +, -), where ‘+?’ and ‘-?’ are used to denote gray zones where it is not clear whether values of state variables are normal or not, so the accuracy and speed of search could be improved.

The SDG approach usually utilizes the qualitative influence between process variables, and depth-first search was mainly used online to determine the fault propagation path

and possible fault origin candidates. The online search proceeds from the effected nodes back to the cause nodes, which makes the diagnosis time become longer when the system becomes bigger. Instead of backward fault diagnosis, Kramer and Palwitsch introduced a rule-based method using SDGs for fault diagnosis [9]. The problems of poor diagnosis resolution and sensitivity to alarm thresholds were discussed in [9]. Using a rule based approach, the processing time was notably reduced. In fact, the processing time for the same test problem formulated by Shiozaki in 1985, containing 99 nodes and 207 arcs, was solved in a few seconds instead of 5 minutes. However, the method has problems in dealing with noisy data and with variables with a compensatory response (CR, meaning they return to its original values) or inverse response (IR, meaning their final qualitative values are opposite to the initial ones). CR/IR occurs when a node under consideration exhibits conflicting behavior due to multiple feed-forward paths from the root node or when a node is in a negative feedback (control or noncontrol) loop. The corresponding variables are called compensatory variables (CVs) and inverse variables (IVs). Whenever a CR/IR is exhibited, earlier methods failed to characterize the ultimate steady state response. To overcome these problems, Oyeleye and Kramer introduced steady-state qualitative simulation [10]. The main goal was to improve diagnosis resolution without loss of completeness. Additional arcs are drawn and an extended SDG (ESDG) is developed to propagate the effect across IVs and CVs. The use of non-causal and causal redundant equations to eliminate spurious solutions was also first proposed in [10].

Chang and Yu proposed another algorithm to overcome several problems associated with SDGs [11]. SDG models were simplified based on process "common sense", and a systematic procedure for rule development based on the SDG have been studied in this work. The continuous system response is divided into several states, and different conditions (truth tables) are used for each state. For multiple propagation

paths, the dominant way is decided based on steady state gains. A special feature of this algorithm is the way it deals with control loops. The variables associated with the integrators are expressed in velocity form. The proposed methodology showed better results on a CSTR model simulation.

In 1994, Wilcox and Himmelblau developed an enhanced SDG approach, called the possible cause and effect graph (PCEG) methodology [12, 13]. PCEG inherits a number of properties from SDGs, such as easy construction, completeness, dealing with cycle systems (systems with loops). Moreover, it overcomes some of the drawbacks of the original SDG methodology. PCEG successfully reduces the search space by providing more accurate information. Other researchers took advantage of PCEGs and approached the fault diagnosis problem using a dynamic probabilistic model, and the time delay problem was also managed [14].

2.3 Attempts to Add Quantitative Information

Due to the qualitative nature of pure SDG models, their low diagnosis resolution restricts their usage. Since the 1990's, the use of fuzzy logic and fuzzy set theory to improve the diagnosis resolution in SDG model-based approaches have been discussed by some researchers. Quantitative information has also been added to the SDG, for better understanding of the dynamic system and spurious fault candidates were reduced dramatically.

The use of fuzzy set theory to help address the problem of alarm threshold sensitivity was discussed by Han et al. [15]. In their approach, after the strongly connected components which are the possible fault origins are located, fuzzy logic is introduced. Based on their membership degree, variables are sequentially arranged and the most probable fault origins are located. This approach was shown to improve the accuracy

of diagnosis resolution. Shih and Lee [16, 17] discussed the removal of spurious solutions using fuzzy logic principles with SDGs, and the Fuzzy Cause-Effect Digraph was proposed. The spurious interpretations attributed to system CR and IR from backward loops and forward paths in the process have been eliminated. Furthermore, this method also can estimate the state of the unmeasured variables, to explain fault propagation paths and to ascertain origins.

The combined use of SDGs and fuzzy reasoning for fault diagnosis was also discussed by Tarifa and Scenna [18]. An SDG is used to model the process to be supervised, and an IF-THEN rule base is compiled, one rule for each potential fault. Fuzzy logic is used in the evaluation to overcome the problems caused by noisy data, CR, IR, and model limitations. The fault whose rule has the highest value of certainty should be the first one considered by the operator. Application to a multi-stage flash desalination plant has been performed using this approach [19].

The literature on combining fuzzy logic and qualitative models looks at improving the representational scope of qualitative models by increasing the granularity through the use of fuzzy representations of real-valued functions. For that reason, Venkatasubramanian et al. believe that these hybrid approaches seem to hold promise [2].

2.4 Recent Activity

SDG based diagnosis approaches have been thoroughly studied by Professor Venkatasubramanian of Purdue University, USA and some of his students. More systematical theory and more comprehensive applications have been developed. The scope ranges from SDG-based fault diagnosis, sensor location determination, operator training, to hazard and operability (HAZOP) analysis, and from single SDG-based methods to hybrid methods. Single fault diagnosis has been studied as well as multiple fault di-

agnosis. In recent years, Maurya et al. have also presented several significant papers regarding fault diagnosis using SDGs.

From their view, SDG-based analysis is based on shallow knowledge (gained from experience) and intuition. Maurya et al., based on earlier contributions of Mylaraswamy et al. [20], have proposed a comprehensive and systematic framework for the development and analysis of SDG-based models and proved feasibility and correctness [21, 22]. Attention also has been paid to the conceptual relationship between the analysis of graph models and the underlying mathematical description of the process [21]. Control loops and flowsheet analysis were, for the first time, discussed thoroughly [22, 23]. The elimination of spurious solutions was achieved by using causal and noncausal redundant equations. Analysis of inverse and compensatory response was also thoroughly discussed. The application of fault diagnosis was proved for small examples as well as a flow-sheet size chemical process in this serial work, and the results are quite promising.

These papers are important contributions to the literature. Some aspects are touched for the first time, for example, a systematic and proven analysis methodology of SDG model is given, and control loops within the SDG framework have been comprehensively addressed. Though the approach proposed is systematical, comprehensive and more objective (all the analysis is based on the manipulation of mathematical equations), the requirements for development is complicated for complex processes. It requires more detailed and precise mathematical equations, which are not easy to get in practice, and the algebraic manipulation for redundant equations is difficult.

After more than 25 years of development, SDG-based approaches are becoming more and more mature. Different approaches have been proposed to deal with different problems, or combined with other approaches to fulfill different requirements. They all showed promise for the various problems solved.

Chapter 3

Development of the Diagnostic System

An efficient diagnosis method based on the Signed Directed Graph (SDG) approach is presented in this thesis, to identify possible causes of process disturbances and faults. It is based on compiling an expert system rule base, using all the known process knowledge to increase the diagnostic resolution.

Figure 3-1 shows the overall scheme of the fault detection and diagnosis system. This method includes two stages, the first one is done off-line to form the rule base expert system, while the second one, fault diagnosis, is carried out in on-line mode. In the off-line stage, an SDG model is used to model the process to be monitored. For each potential fault, possible fault patterns are propagated from this graphic model, and system simulation and process knowledge are combined to determine all possible qualitative patterns for each potential fault, and compiled into IF-THEN rules, one rule for each potential fault. The problems caused by CR and IR are solved by dividing the patterns into initial response and ultimate response. For faults with the same pattern, quantitative data is used to distinguish them from each other. All of

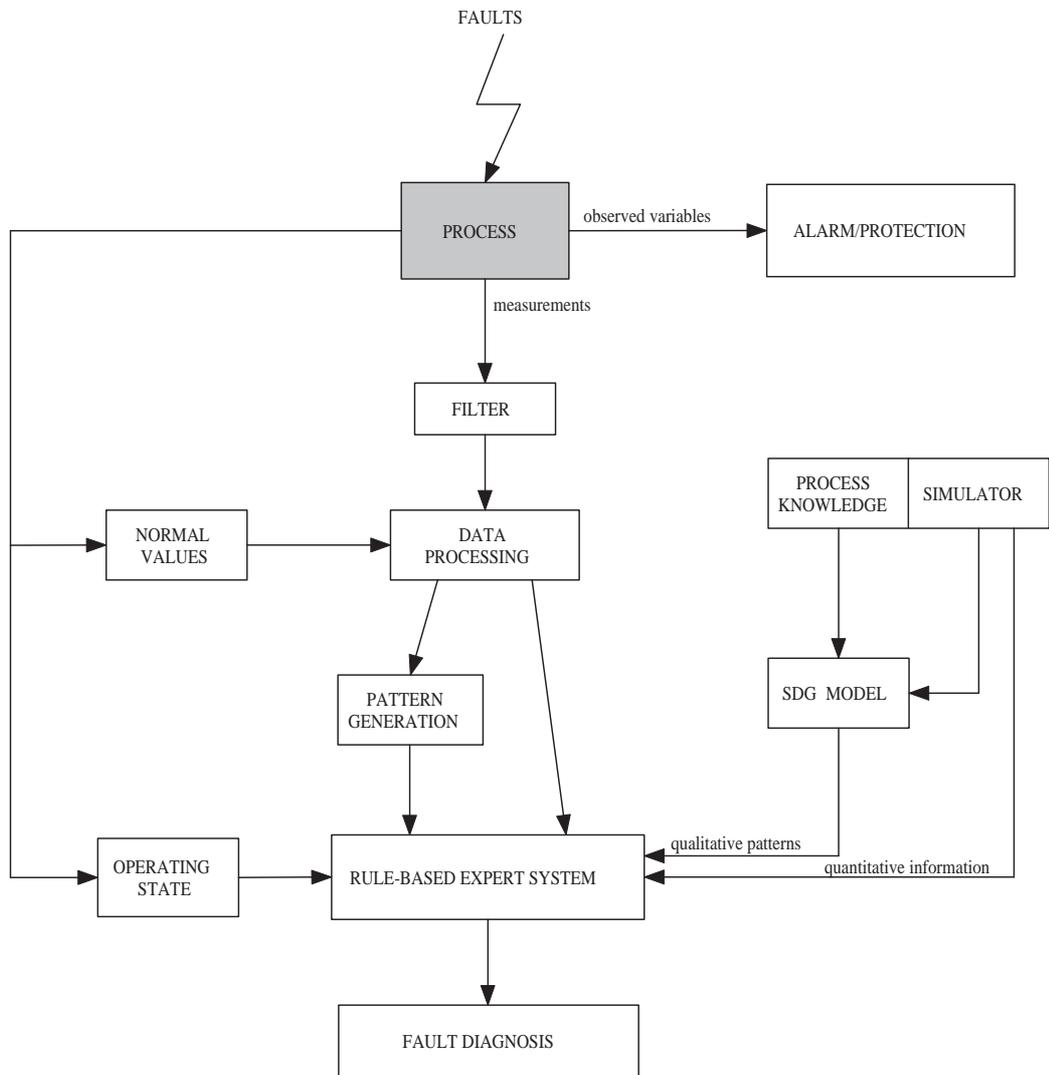


Figure 3-1: Architecture for the Fault Diagnostic System

these compose the rule-based expert system, which is used for online diagnosis.

The online stage is operated in parallel with the monitored process. The FDI system runs all the time during operation, and operation state (such as start up, shut down, or operation point changes) is also an input to the FDI system. In online diagnosis, measurements, after being filtered, are transformed into qualitative patterns by comparing them with the variables' nominal values. The patterns and some needed quantitative data are input into expert system, and matched faults are displayed as fault diagnosis results.

In this algorithm, explicitly calculated variable qualitative values are used for the first time as possible patterns to form the rule base, which increases diagnosis resolution significantly from former works. Explicitly predicted patterns became possible based on the premise that initial responses can be propagated from shortest paths in the SDG and on the use of process knowledge. For each fault, the corresponding rule is composed by initial response patterns and ultimate response patterns, which are both calculated precisely by the help of process knowledge. Initial response patterns make earlier fault detection and diagnosis possible, and ultimate response patterns overcome the problems related to CR and IR, and ensure the correctness of diagnosis results. Because quantitative information and process knowledge are combined, distinguishing faults with the same patterns also becomes possible, and the diagnosis resolution is increased even further.

3.1 Introduction to SDG-based Fault Diagnosis

An SDG is a qualitative model for process diagnosis. The SDG model represents pathways of causality for fault propagation. The nodes of the SDG correspond to state variables, alarm conditions, equipment, or failure origins, and the directed arcs

represent the causal influences between the nodes. Nodes in the SDG assume values of (0), (+) and (-) representing the nominal steady-state value, and higher and lower than the nominal steady-state value, respectively. Arc signs (called arc values) of (+) and (-) indicate whether the cause and effect change is in the same direction or opposite direction. The causal relationships can be constructed from the system topology. Some additional SDG related terms are listed in Appendix A.

The first step of the qualitative approach is to transform the quantitatively precise process measurements into qualitative states of high (+), normal (0), and low (-) [11]. For a variable x , with measured value x_m and normal value x_n , we define

$$d = \frac{x_m - x_n}{\Delta h} \quad (3-1)$$

where Δh is the threshold of the variable. The quantitative variable x is transformed into a qualitative state based on the above equation (3-1). It takes a value of (+), (0), or (-) for $d \geq 1$, $1 > d > -1$, or $d \leq -1$, respectively. For the purpose of this thesis, these three states, when they appear in the pattern, will be represented by (+1), (0), and (-1), respectively.

A small example is given in Figure 3-2 for understanding the concept of an SDG model. This model has five nodes (the node name is the same as the variable name),

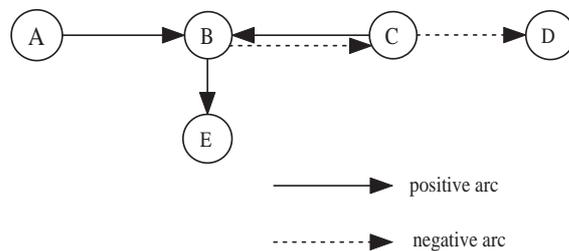


Figure 3-2: Example of SDG Model

and several arcs connecting them; the solid line means that the arc value is (+), and the dashed line mean the arc value is (-). Node A represents an exogenous or

independent input variable. This style will be used for all the SDG models used in this thesis.

In a fault free scenario, all nodes take the value of zero. When a fault happens, the fault node first changes sign, then it will propagate along directed arcs throughout the whole SDG, so some or all system variables will deviate from the normal steady state and the corresponding nodes will change values from 0 to +1 or from 0 to -1. The deviations of the process variables caused by this fault are the fault symptoms, and the set of all the symptoms caused by this fault (a combination of node signs) is the pattern of the fault [5]. For example, in the SDG model in Figure 3-2, if fault node A is (+), then a possible pattern of the fault is $[A(+), B(+), C(-), D(+), E(+)]$, or more concisely $P = [+ , + , - , + , +]$, where the node names are understood.

For fault diagnosis, the procedure involves locating all possible root nodes, given a fault pattern, by propagating backward along consistent branches, from the effect nodes to cause nodes. The fault origin is then located at the maximum strongly connected component (MSCC) in the cause-effect graph. An MSCC could be an strongly connected component (SCC) or a node; for example, node A in Figure 3-2 is an MSCC. For fault simulation, the potential fault origin node first deviates from its normal value, then the fault propagates forward along the pathways. Each directed tree (interpretation), branching from a given root node, forms a pattern for this fault. The procedure goal is to find all possible patterns for the potential fault.

As mentioned by Kramer and Palowitch, the diagnosis problem is the inverse or dual of fault modeling [9]. The process of finding a fault origin from the measurement is basically a form of backward reasoning; on the other hand, the process of fault modeling, or fault simulation, involves forward reasoning, which is more often adopted by most process expert systems. In this thesis the concept of forward reasoning is adopted. As compared with diagnosis, fault modeling is relatively simple. Another

reason for discarding backward reasoning is that for an SDG approach backward online searching is time consuming and less efficient.

Most SDG-based methods assume that there is only a single fault that affects a single node (root node) in the SDG, which is the source of all disturbances. It is also assumed that the fault does not change other causal pathways in the digraph. These two assumptions are also adopted in this thesis.

For a complex chemical process, there are unavoidable cycles and feedback or feed forward loops, either negative or positive. The corresponding digraph is no longer a simple tree, and the propagation path is also complicated. For a simple example, if some nodes in an SDG model are affected by two opposite effects, without sufficient quantitative information, it cannot be determined which is the dominant pathway, so both of them must be considered. For a given digraph and a given fault origin, there may be many interpretations of the fault propagation. Multiple pathways and loops make the propagation more ambiguous. Because of the ambiguity in deciding the qualitative state of a variable, the corresponding resolution of the fault diagnosis algorithm is poor. However, only one or a small set of these interpretations reflects the real behavior of the plant. So how to minimize the interpretation set is the main problem that inhibits increasing diagnosis resolution.

3.2 Building the SDG Model of the System

An SDG model is extracted from the system topology, and characterizes the system behavior in the form of a digraph. SDGs can be obtained either from: (1) plant operation data and/or operator experience, or (2) a mathematical model of the process [5]. In chemical engineering, most process knowledge is well structured and the process mathematical model generally can be written based on material balance and energy

balance. Therefore, it is desirable to construct the SDG from the underlying mathematical model, especially when the operation data and the experience of operators are not sufficient to obtain a consistent representation of the process.

It is believed by some researchers that the process of developing these models is prone to human error and thus unreliable [21, 24]. Therefore systematical methods such as given by [21] and [24] are useful for building an SDG model from the structure of the underlying mathematical description of the system. The method proposed depends heavily on the correctness of the mathematical model, and the nodes are all system variables which appear in the mathematical models. Actually, a complete SDG model also contains some nodes, like equipment, fault origins, etc. which are not systems variables, and could not be included without operator experience or operation data. Therefore, an SDG model can be constructed based on both types of knowledge, to make it more correct and complete.

Constructing an SDG model is not a central part of the thesis. Here just a general idea of SDG model building from the underlying mathematical model is presented. In general, a system can be described by a set of differential equations (DEs) or algebraic equations (AEs), or a combination of DEs and AEs, called a DAE system, written in the following form:

$$dx_i/dt = f_i(x_1, x_2, \dots, x_n) \quad (3-2)$$

$$x_i = \sum_{j=1, j \neq i}^n a_{ij}x_j \quad (3-3)$$

All the variables x in the equations are nodes in the digraph. Directed arcs are drawn from all of the variables on the right hand side to the system variables on the left hand side in the equations. For arcs from ordinary differential equations, the arc

values are determined by:

$$\text{sign}(x_j \rightarrow x_i) = \text{sign}(\partial f_i / \partial x_j) \quad (3-4)$$

if $\partial f_i / \partial x_j \neq 0$, or if it is zero, then it should be replaced by $\partial^m f_i / \partial x_j^m$ if m is odd, and by $\partial^m f_i / \partial x_j^m [dx_j]$, if m is even, where m is the order of the first non zero partial derivative [10]. For algebraic equations, the arc values are determined by:

$$\text{sign}(x_j \rightarrow x_i) = \text{sign}(a_{ij}) \quad (3-5)$$

Maurya et al. [21] presented a comprehensive and systematic framework for the development and analysis of SDG-based models and proved the feasibility and correctness of this approach for SDG model building and analysis for system equations in DE, AE, and DAE form. Note that for systems described by DEs, there is explicit causality from the variables on the right hand side to the variable on the left hand side, but for systems described by AEs, there is no causality, the system only captures instantaneous behavior.

All the variables in the mathematical model are shown in the SDG model, including measured and unmeasured variables. For measured variables, the working sensor is lumped with its corresponding variable node. Thus a sensor will not be shown explicitly in the SDG model, except for the sensor of a controlled variable which is a potential root node, for which a separate node is given. The sensor being lumped with its corresponding variable is based on the assumption that the measurement is instantaneous, so it cannot generate spurious or erroneous interpretations.

Alarm conditions (maximum or minimum value), sensors or actuators in control loops (if they are potential fault nodes), and other possible fault origins also can be shown in the SDG model according to the need for fault analysis and determining fault sources.

These nodes and arcs can be drawn based on process knowledge and experience. For example, sensor and actuator nodes in a control loop can be drawn as shown in Figure 3-3: The variable B is the controlled variable, the sensor node variable is denoted by

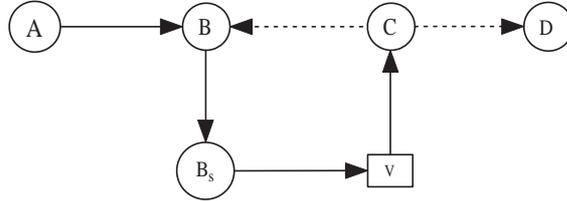


Figure 3-3: SDG for a Control Loop

the subscript ‘s’, and the valve node is enclosed in a rectangle instead of a circle. As shown in the figure, it is easy to determine arc directions and arc values between these nodes according to experience.

3.3 Form of the Diagnosis Knowledge Base

The development of this thesis is based on the fundamental structure of Maurya et al. [21]. The algorithm proposed here is also based on the single fault origin assumption, which means there is only one cause of any specific fault. From the view point of engineering, this is reasonable since multiple faults seldom appear at the same time [25].

Using a rule-based approach for fault diagnosis was first proposed by Kramer and Palowitch [9]. They remarked that qualitative simulation is a combinatorial explosive problem, and to avoid the explosion, they proposed using a rule to explicitly represent the combined set of interpretations of each fault instead of enumerating each interpretation. The same concept was used by Chang [11]. For each fault, a logical statement (rule) is derived from the process digraph. All possible behaviors of the real system are included, through the interpretations related to discrete choices

of the dominant causal pathways. Logical truth tables are used to generate the rules, which combine all the simulation trees (interpretations) on the order of events and the direction of deviation of each node connected to the fault origin.

Rule-based methods improved the efficiency of fault diagnosis. However, in the work of Kramer and Palowitch [9], the assumption of single state transitions was made, which is a severe limitation in the real system, and special cases arise when the unmeasured nodes are removed. Compensatory responses (CRs) and inverse responses (IR) were also not considered. An ESDG approach was proposed by Oyeleye and Kramer [10] to solve this problem, but this requires that new artificial arcs must be added. Chang et al. [11] proposed a way to represent a continuous process response into several discrete states, and imposed different conditions (truth tables) to derive the rules. The improvement is made at the expense of memory size. Moreover, the rules composed by them do not clearly reveal the fault pattern, and the interpretation's structure is lost. In this work, all interpretations are obtained and expressed as patterns composed by all the measured variables. The risk of combinatorial explosions is reduced here by finding the dominant pathways and dividing the analysis into transient response and final response.

It was pointed out by Kramer and Palowich [9] that the SDG model derived from process equations has certain limitations. The most significant one is that the correct diagnosis can be guaranteed only if each variable undergoes no more than one transition between qualitative states during fault propagation. They also made the assumption of single state transition in deriving the rule base format. That method excludes the ultimate response in cases where variables exhibit inverse responses (IR) and compensatory responses (CR), since IR and CR may cause qualitative changes more than once during the propagation of the fault. The approach in this thesis overcomes this limitation.

For qualitative simulation, there are only three qualitative states, normal (0), high (+1), and low (-1). It is assumed that the system process is at steady state before fault initiation, so all nodes initially have the value 0. When a fault appears, the deviation of the root node causes all variables accessible from the root node to change sign. The first sign change (0 to +1 or 0 to -1) is the initial response of the variable. Due to multiple feed forward paths and negative feedback paths, the qualitative state of the variable may change again during the propagation of the fault. However, the possible final or ultimate response of the node can only fall into three cases (oscillatory behavior is not considered in this thesis): it may keep the same sign, be compensated to zero (CR), or change to the opposite sign (IR). So, if the initial response of all the system variables can be determined, then the final response can be determined, based on the identification of IVs and CVs.

Between initial response and final response, there is usually an intermediate transient period. The response corresponding to this period is sometimes complex, and may not provide any insight. However, unlike quantitative analysis, in which the precise measurements keep changing for every sample during the transient period, the qualitative states of system variables will keep the initial sign change for a time before they possibly change sign again. That means the initial response patterns would remain for a while. This observation has been used to advantage in the algorithm proposed in this thesis. The significant benefit is that information about time delays and time constants need not be considered.

The whole rule base has three parts, corresponding to the normal state before a fault appears, the initial response and the ultimate response after the fault occurs, respectively. Each potential fault has a corresponding rule for both responses. Each rule consists of qualitative patterns propagated from the SDG model and quantitative rules extracted from quantitative information, which will be discussed later.

Each pattern is a combination of the qualitative state of all the measurements at the same sample time. For example, for a system with 5 measured variables, the pattern $P = [0, 0, 0, 0, 0]$ is the Normal Pattern at nominal steady state; when there is a fault the possible patterns may be $P = [1, 1, -1, -1, -1]$ in initial response and $P = [0, 1, 0, -1, -1]$ in ultimate response.

An example rule can be written as:

IF $P = [0, 0, 0, 0, 0]$ THEN the system is normal, or

IF $P = [1, 1, -1, -1, -1]$ or

IF $P = [0, 1, 0, -1, -1]$ THEN A is high.

The fault is isolated as it is diagnosed. It should be noticed that at either the initial or ultimate response stage, there may be more than one pattern appearing in the rules. The rules compiled in this thesis are the enumerated interpretations, not just logic expressions as shown in other works; the patterns explicitly show the qualitative state of each node. The effects of IR and CR are shown in the final response patterns.

The rule base does not include any consideration of specific response time, just as initial, transient and ultimate response. Generally, the initial sign change of a node will be maintained for a few sample times during the transient period before it arrives at another steady state. However, the time from initial pattern change to final pattern is not explicitly calculated in the rule base. The proper sequence can be achieved by firing the final response rule set several seconds or minutes (depending on the system time constants) after the initial rule set has been fired, as shown in Figure 3-4. The time counter is not easy to program on a PC, but it would not be a problem in industry. In fact, because of control loops, some variables in the ultimate response will return to zero instead of maintaining the deviations seen in initial response, and the difference is obvious, so in the next chapter on implementation, the pattern change

time is not considered.

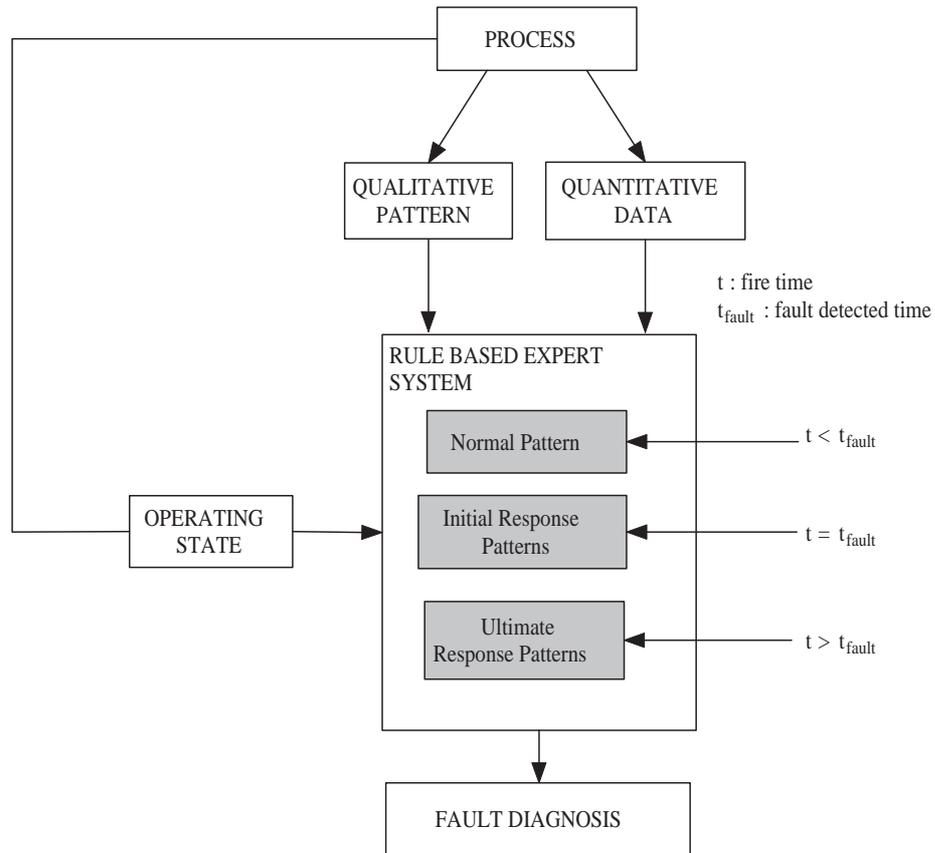


Figure 3-4: Implementation of the Algorithm

3.4 Deriving Patterns from an SDG

Given an SDG model, in a fault free situation, all nodes take value of zero. If any node has a nonzero value, the system is in a failure state. When a fault happens, the deviation of the root node is the source of all subsequent changes, and the system variables which lie on a directed path will deviate from the normal steady state, changing from 0 to +1 or to -1. All measured variables are used to form the pattern. It should be noticed that all process variables that are not accessible from the root node must be normal (0) under the single fault assumption. So, if the pattern is

composed of all the measured variables it is possible for some of them are zero in the initial response pattern. Unmeasured variables are not removed from the SDG in propagation (removing unmeasured variables may cause erroneous interpretations).

In section 3.3, it is pointed out that if the initial response of all the system variables can be determined, then, the final response can be determined. Theoretically, the initial response of a system variable x_j due to changes in an exogenous variable e_l , can be predicted by propagation through all of the shortest path(s) from e_l to x_j in the SDG, and the final response can be predicted by propagation through all of the directed paths. However, if the initial response pattern of all the variables can be determined precisely, and the nodes corresponding to IVs and CVs are identified, then for the final response pattern, we only need to change the corresponding signs of IVs and CVs. This approach is novel to this thesis.

3.4.1 Initial Response

Claim: The initial response of a system variable x_j due to changes in an exogenous variable e_l , can be predicted correctly by propagation through all of the shortest path(s) from e_l to x_j in the SDG for DE systems and DAE systems with only one perfect matching, i.e., the corresponding SDG is unique [21]. Note that the arc lengths in the SDG of a DAE system are determined as follows: The arcs in the DE part of the SDG have length 1 and the arcs in the AE part of the SDG have arc length 0 because of the instantaneous response behavior [21].

The arc length of a control loop is not discussed in [21]. However, it is observed that a PI controller effectively behaves as a P controller in the initial response of the system, and an integrator does not play an immediate role. A P controller acts as an AE system, so in this thesis, it is defined that the arc length from the variable

measurement to the manipulated variable is 0 for initial response.

Spurious interpretations may be generated because of the ambiguity of the SDG in predicting the fault propagation pathways. For a given digraph and a given fault origin, among many interpretations of the fault propagation, only one or a small set of these interpretations reflects the real behavior of the plant. This means that among all the propagation pathways branching from the given fault node, there are dominant ones which are the real interpretations of the fault. The dominant pathways can be obtained by process knowledge and process simulation.

The simulations of different magnitudes of a fault can provide the information of sign change directions of the measured variables and the order of events. This knowledge also can be used to identify the dominant pathways in the digraph. Former researchers have used this approach [9, 10, 11, 26]. It is believed that incomplete knowledge leads to limited diagnosis capabilities, and SDG analysis would be incomplete without process knowledge. Process knowledge includes information about the process, a process model, history data, and heuristics [27].

Under the above claim [21] the initial response for the entire system will be much less ambiguous than propagation through all of the directed paths. In some cases, the initial response for all the variables is unambiguous. However, in some cases, if there are more than one shortest path, spurious or erroneous interpretations will be generated if the paths have the opposite sign. Under this condition, dominant paths need to be determined by combining process knowledge and simulation analysis, and the initial response can be determined by the dominant shortest path. All the qualitative states are thus determined for the initial fault pattern.

3.4.2 Ultimate Response

Oyeleye and Kramer [10] have contributed important ideas on steady state analysis. The necessary conditions to identify IV and CV (with respect to a local exogenous variable), have been given in detail, with explanation and proof. Detailed conditions for IV and RV are listed in Appendix B.

Generally, IV/CV should be located inside a SCC in a negative feedback loop (cycle) [10]. In this thesis, each IV/CV is first identified by using the necessary conditions, then system analysis and simulation will be combined for understanding the location of IV/CV. After the identification of IVs and CVs with respect to local exogenous variables, the final response fault pattern is obtained just by changing the corresponding variable signs.

It should be pointed out that the CR of controlled variables in feedback control loops is a special case of the more general behavior of CVs in negative feedback loops [10]. In chemical process systems, the design purpose of a feedback control loop is to attempt to maintain important variables within an acceptable range in the presence of most expected faults and disturbances. Thus, in most cases, the controlled variables should maintain zero deviation after all transients have died out and a new steady state is reached. However, the control loops may be saturated because of the large size of a fault, or the fault may occur inside the control loop; under both conditions, it would not work properly and the system cannot return to its nominal state. Thus, in the ultimate response pattern, the controlled variable will become zero if the control loop works properly, otherwise variables will keep the same sign as initial response.

3.5 Combining Quantitative Data

Since an SDG-based approach focuses on qualitative information of the process, diagnostic limitations are unavoidable and increasing resolution becomes difficult. The IF-THEN rule for each fault is composed of several patterns corresponding to different possible states or interpretations. For multiple paths between variables, the dominant path can be decided easily by using process knowledge and experience. Using a simulation model will help verify each fault pattern, and modification can be made for each rule accordingly. So the rule for each fault has been minimized as far as possible using qualitative data. However, there still may be rules with the same pattern.

For those faults with the same patterns, further quantitative knowledge can be used to form additional parts of rule base. If two faults have the same pattern, it implies that these faults have affected the same variables in same directions. However, if quantitative information is considered, differences may be seen. For different faults, variables may deviate in the same direction, but that does not mean that they deviate from their normal values in the same way.

Usually, faults with the same pattern are pairs of disturbances and sensor faults, which affect the same variables inside a control loop. Sensor failures in control loops can be distinguished from actual process malfunctions because of the different action of control loops. The simulation results also demonstrate this observation.

The sensor faults considered in this thesis are step biases. Such a fault can be isolated from other malfunctions if one more piece of process knowledge is used. Since all the variable values are read from their respective sensors, so a step sensor fault means that the sensors fault produces an instantaneous change in the corresponding variables, but changes such as a disturbance do not, they produce a slower response, because

the dynamic variables need time to respond.

So suppose a fault happens at time sample k , sensor values at points $k + 1$ and $k + 2$ are different for these two types of fault. For a sensor fault, there is an instantaneous change, so the value will go directly up or down to a maximum value at the $k + 1$ point, and then go back to a new steady state. so if V is the variable value, for a sensor fault the $|V(k + 1)|$ is greater than $|V(k + 2)|$; but for a disturbance, $|V(k + 1)|$ is less than $|V(k + 2)|$. So only using these two points we can readily isolate these two faults.

For example, assume that sensor fault A(+) has the same pattern as disturbance B(+), that node V is the direct descendant of both nodes, and V takes value (+). Generally, the complete rule for fault A (shown previously in Section 3.3) can be extracted as:

IF $|V(k + 1)| > |V(k + 2)|$, and $P = [1, 1, -1, -1, -1]$ THEN A is high.

There is small chance of error for fast sampling, i.e., the interval between two sampling points is too short, however, this can be solved by looking at $V(k+2)$ and $V(k+4)$. Or, if we calculate the gradient between sampling points, the one close to rectangular will be sensor fault. This provides another example of the importance of considering process knowledge, even if it is case specific.

3.6 Online Fault Detection and Diagnosis

As shown in the architecture schematic (Figure 3-1) at the beginning of this chapter, measurements, which may be data from a Distributed Control System (DCS), are first processed, including noise filtering, then measurements are compared with nominal steady state values, and finally the processed data is transformed into qualitative

values, either (+), (0), or (-). For rules compiled from quantitative information, corresponding data is also input to the expert system. This is the second step.

The third step is pattern matching and online diagnosis of the fault origin. When a fault occurs, the corresponding system variables will change (from 0 to +1 or -1). The pattern values are collected and compared with the knowledge base; the fault for which the pattern matches one of the rules is the root fault, and the result will be displayed. The fault pattern for the initial response fulfills the requirement for a quick diagnosis, if it is unambiguous; otherwise the final response can be used to ensure the diagnosis result correctness.

3.7 Fault Detection

Limit-value checking is used in this thesis for fault detection. The classical limit-value checking for monitoring and fault detection is still widely used in most process industries, mainly because of its simplicity and reliability. Within automatic process control systems, measurable variables are monitored, checked with regard to tolerance, and alarms are generated for operators when variables deviate significantly from the nominal value. For a dangerous process state, the monitoring function automatically initiates an appropriate response. At the same time as the alarm is raised, the fault diagnosis system is triggered to perform FDD and the results are shown to operators who can take the corresponding action in order to maintain operation and to avoid damage or accidents.

When a fault happens, it should be detected as early as possible. This can be done by detection of the fault symptoms. Using SDG-based fault diagnosis, in a fault free scenario, all nodes take a value of zero. If any node has a nonzero sign, there is a fault in the system. For the algorithm presented here, the fault detection is not a

separate part, instead, it is integrated into fault diagnosis. The fault can be identified almost at the same time as it is detected, if time delays and time constants are not large. A limitation of limit-value detection is that it does not perform well during transients or large fluctuations; additional fault detection approaches may be needed at this time.

The qualitative state of each process variable can take the value normal (0), high (+), or low (-), by comparing it with the nominal steady state value. Due to uncertainty and noise in the data measurement, the qualitative states are obtained considering an appropriate threshold of each variables. For example, if the measurement of a variable is above the upper limit alarm value, then it takes +1; if it is lower than the lower limit alarm value, then it takes -1. It is easy to transform variable measurements to qualitative values via comparing with its threshold. On the other hand, the setting of a threshold should also reflect the qualitative process knowledge. For example, for measurements in the neighborhood of a threshold, a small change of the threshold may change the qualitative value assigned to the measurements.

Compromises have to be made setting thresholds, between detection sensitivity and false alarms due to normal fluctuations. When an abnormal variable is in the neighborhood of the designed threshold, SDGs have difficulty providing an accurate resolution. In principle, expert systems approaches are amenable to fuzzy logic, which may help address the problem of alarm threshold sensitivity [9]. This is a topic for future research. However, the diagnostic algorithm proposed here is based on a set of logic rules, which can be combined with other rules pertaining to plant operations in an expert system. In this sense, this thesis will be of significant benefit for future implementations.

3.8 Conclusion

In this chapter, an intelligent fault diagnosis algorithm has been proposed, along with fault detection and process implementation. The fault diagnosis algorithm combines process knowledge and quantitative information into the SDG approach, improving diagnosis resolution, and facilitating the early detection and diagnosis of process faults. Moreover, using IF-THEN rules to build the fault diagnosis knowledge base makes the diagnosis and isolation process more efficient and allows it to be embedded or combined with other expert systems in the process industries. This is illustrated in a detailed example in the next chapter.

Chapter 4

Implementation and Simulation

Results

4.1 System Model

4.1.1 JCSTR Dynamic Model

The test study has been done on a jacketed continuously-stirred tank reactor (JCSTR) model, shown in Figure 4-1. The tank inlet stream is received from another process unit. The objective is to control the temperature and volume in the tank at desired values via the temperature control (TC) and level control (LC) loops, respectively. A heat transfer fluid is circulated through a jacket to heat the fluid in the tank. The volume is controlled by adjusting the tank outflow, and the temperature is controlled by adjusting the heating fluid inflow valve. In this model, we assume that no change of phase occurs in either the tank fluid or the jacket fluid. The jacket volume, density, and heat capacity of liquids are assumed to be constant.

Under these assumptions, the closed loop system model is governed by equations 4-1

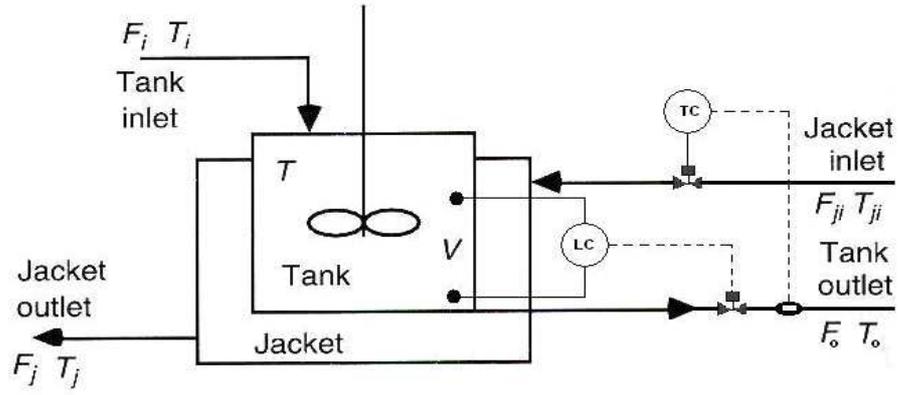


Figure 4-1: JCSTR Model and Controller

to 4-8. Names and values of parameters related to the JCSTR and normal values of variables are given in Table 4.1.

$$\frac{dV}{dt} = f_1(F_i, F_o) = F_i - F_o \quad (4-1)$$

$$\frac{dT}{dt} = f_2(F_i, T_i, A, V, T_j) = \frac{F_i(T_i - T)}{V} + \frac{UA(T_j - T)}{V\rho c_p} \quad (4-2)$$

$$\frac{dT_j}{dt} = f_3(F_j, T_{ji}, A, T) = \frac{F_j(T_{ji} - T_j)}{V_j} - \frac{UA(T_j - T)}{V_j\rho_j c_{pj}} \quad (4-3)$$

$$A = \frac{\pi(D_r)^2}{4} + \frac{4V}{D_r} \quad (4-4)$$

$$e_V = V_{sp} - V \quad (4-5)$$

$$e_T = T_{sp} - T \quad (4-6)$$

$$F_o = F_{oss} + K_{P1}e_V + K_{I1} \int e_V dt \quad (4-7)$$

$$F_{ji} = F_{jiss} + K_{P2}e_T + K_{I2} \int e_T dt \quad (4-8)$$

Parameters	Meaning	Value
D_r	Diameter of the reactor (m)	5
c_p, c_{pj}	Heat capacity ($j/kg.K$)	4.19 *1000
U	Heat transfer coefficient (W/m^2K)	852
ρ, ρ_j	Density (kg/m^3)	997.95
A	Area for heat transfer (m^2)	163.64
V	Mixture volume (m^3)	180
V_j	Heating water volume (m^3)	9
T	Tank temperature (K)	33.6 + 273
T_j	Jacket temperature (K)	104.3 + 273
T_i	Temperature of the mixture feed (K)	10 + 273
T_{ji}	Temperature of the heating water feed (K)	120 + 273
F_i	Mixture inflow (m^3/s)	0.1
F_o	Mixture outflow (m^3/s)	0.1
F_{ji}	Heating water inflow (m^3/s)	0.15
F_j	Heating water outflow (m^3/s)	0.15
F_{jiss}	Heating water inflow steady state value (m^3/s)	0.15
F_{oss}	Mix outflow steady state value (m^3/s)	0.1
V_{sp}	Volume set point (m^3)	180
T_{sp}	Temperature set point (K)	33.6 + 273
K_{P1}	Volume proportional gain	0.0024
K_{I1}	Volume integral gain	1.46e-6
K_{P2}	Temperature proportional gain	0.033
K_{I2}	Temperature integral gain	4.6e-5

Table 4.1: Parameters and Variables of JCSTR Model

4.1.2 JCSTR SDG Model

The corresponding SDG model built according to the DAE equations and control principle is shown in Figure 4-2.

Directed arcs are drawn from all of the variables on the right hand side to the system variables on the left hand side in the DAE equations. For arcs from ordinary differential equations, arc values are determined by Equation 3-2. Three DEs are given in Equations 4-1 to 4-3. The calculation of arc values is given in Table 4.2. Signs of most arcs can be fixed without using any numerical information, and only the arc

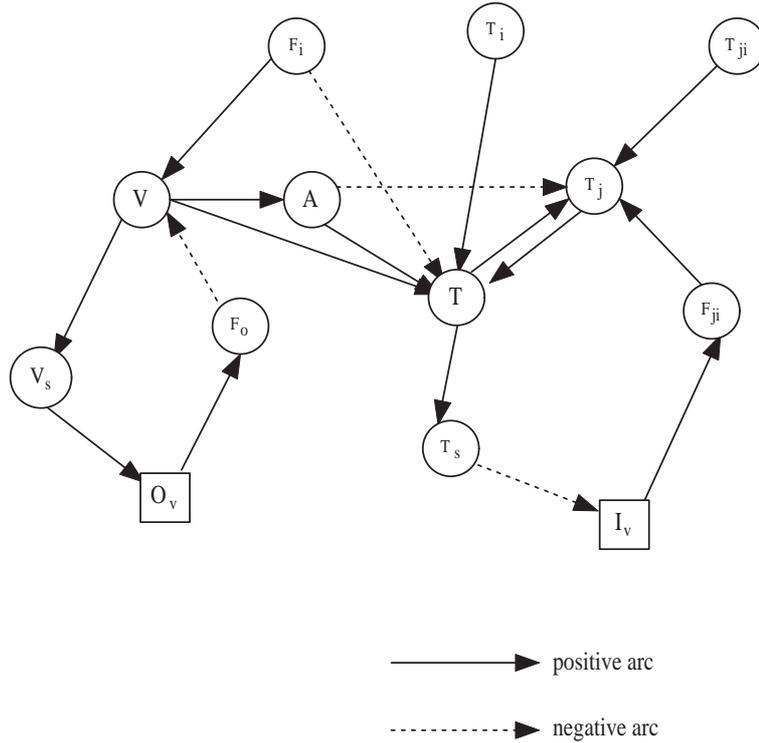


Figure 4-2: SDG Model for JCSTR Model

$(V \longrightarrow T)$ requires some numerical information.

For algebraic equations, the arc values are determined by Equation 3-3. Thus for arc value of $V \longrightarrow A$ by the AE equation 4-4 is $a = \frac{4}{D_r} > 0$.

In this SDG model, the tank temperature and volume control loops are connected by adding sensor nodes and valve position nodes according to the control principle mentioned previously. Notice that the SDG constructed is rather robust, since changes in the parameters generally will not affect the graph.

The notation is consistent with the last chapter, where the node variables are the same as in the system model, sensor nodes are denoted by the subscript 's', and the valve nodes are enclosed in a rectangle. I_v is heating fluid inflow valve and O_v is tank outflow valve. The solid line means that the arc value is (+), and the dashed line means the arc value is (-). The arc notation will be the same throughout the

Arc	Arc value calculation
$F_i \longrightarrow V$	$\frac{\partial f_1}{\partial F_i} = 1 > 0$
$F_o \longrightarrow V$	$\frac{\partial f_1}{\partial F_o} = -1 < 0$
$F_i \longrightarrow T$	$\frac{\partial f_2}{\partial F_i} = \frac{(T_{is}-T_s)}{V_s} < 0$
$V \longrightarrow T$	$\frac{\partial f_2}{\partial V} = -\frac{F_i(T_i-T)}{V^2} - \frac{UA(T_j-T)}{V^2\rho c_p} > 0$
$T_i \longrightarrow T$	$\frac{\partial f_2}{\partial T_i} = \frac{F_{is}}{V_s} > 0$
$A \longrightarrow T$	$\frac{\partial f_2}{\partial A} = \frac{U(T_j-T)}{V\rho c_p} > 0$
$T_j \longrightarrow T$	$\frac{\partial f_2}{\partial T_j} = \frac{UA}{V\rho c_p} > 0$
$F_{ji} \longrightarrow T_j$	$\frac{\partial f_3}{\partial F_j} = \frac{T_{ji}-T_j}{V_j} > 0$
$T_{ji} \longrightarrow T_j$	$\frac{\partial f_3}{\partial T_{ji}} = \frac{F_j}{V_j} > 0$
$T \longrightarrow T_j$	$\frac{\partial f_3}{\partial T} = \frac{UA}{V_j\rho_j c_{pj}} > 0$
$A \longrightarrow T_j$	$\frac{\partial f_3}{\partial A} = -\frac{U(T_j-T)}{V_j\rho_j c_{pj}} < 0$

Table 4.2: Arc Value Calculation for DE Equations

thesis, so the explanation will be omitted. The SDG model gives a clear picture of the JCSTR system. The system behavior is visualized by the cause-effect relationships between variables. Volume and temperature control loops shown in the SDG model are also in cycle or loop path form; loop (V, V_s, O_v, F_o, V) and loop $(T, T_s, I_v, F_{ji}, T_j, T)$. These two loops are coupled through the change of heat transfer area when the volume changes.

4.2 Fault Analysis

A simulation model built by Atalla Sayda [28] is used. For this simulation model, potential faults include disturbances caused by independent variable changes, and sensor or actuator faults inside control loops. Totally 10 different faults and disturbances have been selected and studied. These ten faults are listed in table 4.3.

There are five measured variables, as shown in Table 4.4, and all of them will be used

Symbol	Fault Origin
$F_i(-)$	Low Mix Inflow
$F_i(+)$	High Mix Inflow
$T_i(-)$	Low Inlet Temp
$T_i(+)$	High Inlet Temp
$T_{ji}(-)$	Low Heating Fluid Temp
$T_{ji}(+)$	High Heating Fluid Temp
$T_s(-)$	Faulty Temp Sensor
$V_s(-)$	Faulty Volume Sensor
$O_v(-)$	Faulty Outflow Valve
$I_v(-)$	Faulty Heating Fluid Inflow Valve

Table 4.3: Fault Origin List

to compose the patterns. The value is taken from the reading of its corresponding sensor, so the values in the patterns are the values of sensor variables. In the SDG model, because we suppose that the measurement is rapid, the sensors of F_o , T_j , and F_{ji} are lumped with their corresponding variable nodes, except the measurements of the controlled variables V , and T , which are potential root nodes. In the following reduced SDG models corresponding to each fault, if a sensor is not a root node, it will also be lumped with its corresponding variable node.

Symbol	Measurement
V or V_s	Tank volume
F_o	Tank outlet flow rate
T or T_s	Tank temperature
T_j	Jacket temperature
F_{ji}	Jacket inlet flow rate

Table 4.4: Measurements for On-line Diagnosis

4.2.1 Disturbance Caused by Mix Inflow Rate F_i

The SDG model corresponding to a fault in F_i is reduced from the original SDG model, as shown in Figure 4-3. Based on the single fault assumption, there can be only one fault origin in each case, and the other fault origins are deleted in this SDG model, along with their arcs to the system. Two different faults caused by origin F_i are high inflow rate $F_i(+)$ and low inflow rate $F_i(-)$. The following analysis is based on the fault high inflow rate $F_i(+)$. For fault origin mix inflow low, $F_i(-)$, because only the fault origin changes its qualitative state, the propagation paths are the same, so the corresponding fault pattern will be opposite to that of $F_i(+)$. A similar argument may be used for fault origins T_i and T_{ji} .

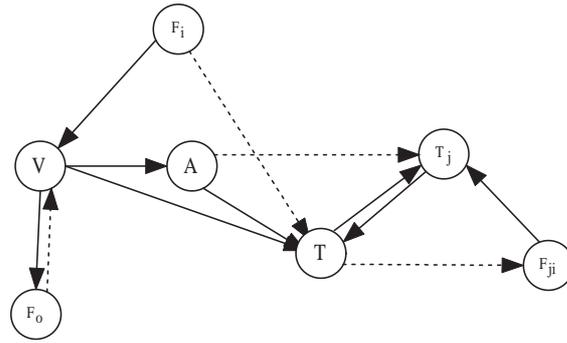


Figure 4-3: Reduced SDG Model for Mix Inflow Fault

The effect of a fault is propagated from the root node (disturbance/fault node) to the nodes representing system variables in the SDG model. The value of each effected node is equal to the product of the cause node value with the arc value, this is consistent with the truth table in Appendix A (Table A-1). For example, fault $F_i(+)$ means the root node $F_i = +1$, the positive arc from F_i to V makes $V = +1$. The purpose of propagation through the SDG model is to find the possible initial response patterns and final response patterns corresponding to the fault $F_i(+)$. The elements of the pattern used in the fault diagnosis for this JCSTR model correspond to the five measured variables $[V, F_o, T, T_j, F_{ji}]$. So the initial response and final response

of the system is found by determining the possible initial response and final response of these five variables for each fault.

4.2.1.1 Initial Response Fault Patterns for $F_i(+)$

As claimed in Chapter 3, the initial response of a system variable x_j due to changes in an exogenous variable e_i , can be predicted by propagation through all of the shortest path(s) from e_i to x_j in the SDG; this is correct for DE systems and DAE systems with only one perfect matching [21]. The JCSTR mathematical model only has one algebraic equation, so it is a DAE system with only one perfect matching, and the SDG model is unique. The arc lengths corresponding to the initial response of the complete JCSTR system is shown in Figure 4-4.

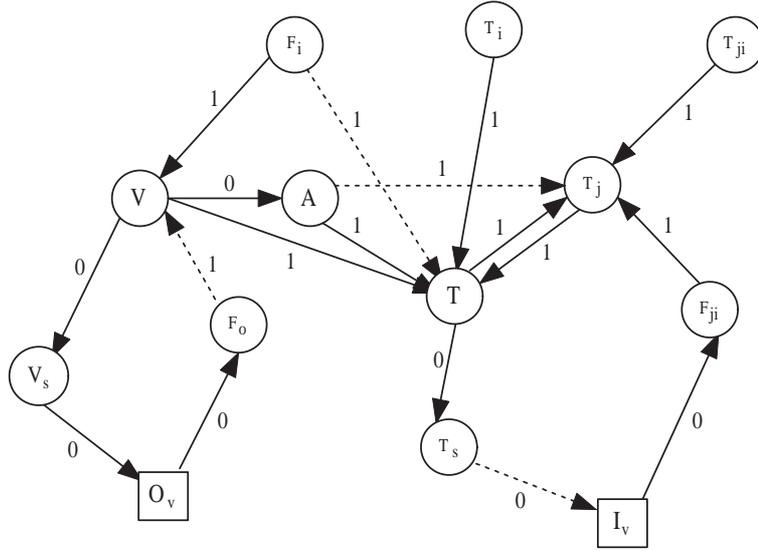


Figure 4-4: SDG Model with Arc Lengths

The initial response of the five variables $[V, F_o, T, T_j, F_j]$, are propagated from the origin node F_i , as follows:

- The shortest path from F_i to V is $F_i \longrightarrow V$ with length 1, which makes $V = +1$.

- The shortest path from F_i to F_o is $F_i \longrightarrow V \longrightarrow F_o$ with length 1, which makes $F_o = +1$.
- The shortest path from F_i to T is $F_i \longrightarrow T$ with length 1, which makes $T = -1$.
- The shortest path from F_i to F_{ji} is $F_i \longrightarrow T \longrightarrow F_{ji}$ with length 1, which makes $F_{ji} = +1$.
- The shortest paths from F_i to T_j are three: path $F_i \longrightarrow T \longrightarrow T_j$ and path $F_i \longrightarrow V \longrightarrow A \longrightarrow T_j$ makes $T_j = -1$, but $F_i \longrightarrow T \longrightarrow F_{ji} \longrightarrow T_j$, also with length 2, makes $T_j = +1$.

The initial response of T_j corresponding to a disturbance of F_i cannot be determined only by the SDG model. Based on the simulation result of fault F_i high, see Figure 4-5, it shows that the initial response of T_j is $+1$, which means the dominant pathway is $F_i \longrightarrow T \longrightarrow F_{ji} \longrightarrow T_j$. In this process, because of the large volume of tank,

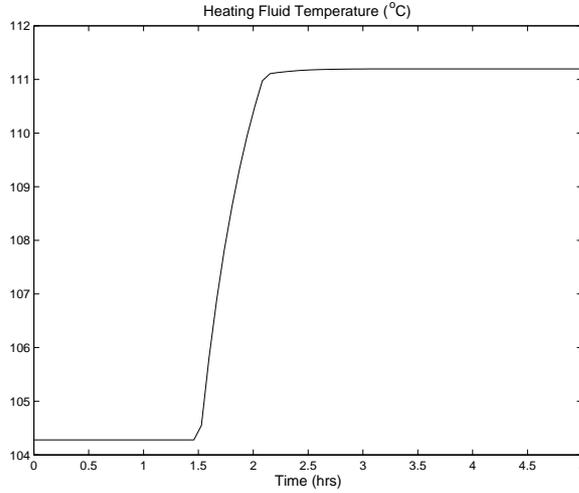


Figure 4-5: T_j Response when F_i High, $t_{0fault} = 1.5$

the heat transfer area A change would be small; the nominal state value of T and T_j are 33.6 and 104.3, and T_j also affects T at the same time, so the influence from T to T_j is likely to be small. It also can be understood easily if we consider the role of

the temperature control loop: because T deviates opposite to the change of F_i , the control loop must respond further to compensate this effect. So the weight of the third path is heavier than the first two, and the initial response of T_j is $+1$. By either approach, process knowledge resolves the ambiguity.

So, the initial response fault pattern P , which is composed by the five variables $[V, F_o, T, T_j, F_{ji}]$, for disturbance F_i high is $P = [+1, +1, -1, +1, +1]$.

4.2.1.2 Ultimate Response Fault Patterns for $F_i(+)$

At steady state or the ultimate response state, there are only three cases: the nodes can be compensated back to the normal state (CVs) or change to an inverse state (IVs), otherwise it has to stay in same as in the initial response.

There are two negative feedback control loops in the SDG model. One is the volume control loop, in which tank volume V is the controlled variable, tank outflow rate F_o is the manipulated variable. The other is the temperature control loop, in which tank temperature T is the controlled variable, and jacket inflow rate F_{ji} is the manipulated variable. Process control loops are usually designed to achieve their intended function, using PI control to attempt to maintain the controlled variables in the desired steady state, in the presence of most expected disturbances. So in many situations zero deviation of the controlled variables is maintained after all transients have died out.

As mentioned in last chapter, CR of controlled variables in feedback control loops is a special case of the more general behavior of CV's in negative feedback loops. With respect to the disturbance F_i , it is easy to identify that V and T are CVs. Thus, one of the possible ultimate response fault pattern for disturbance F_i high, is $P = [0, +1, 0, +1, +1]$, in which both controlled variables V and T return to zero deviation. There are possible exceptions. If the disturbance magnitude is large enough

to cause loop saturation, one or both of them will keep the initial response value, so the other possible fault patterns at the final stage could be: $P = [+1, +1, 0, +1, +1]$, $P = [0, +1, -1, +1, +1]$, and $P = [+1, +1, -1, +1, +1]$, i.e., V is not a CV, T is not a CV, or neither T nor V are CVs, respectively.

4.2.1.3 Fault Patterns for $F_i(-)$

The fault patterns of $F_i(-)$ are opposite to the patterns of $F_i(+)$. So, the initial response fault pattern is: $P = [-1, -1, +1, -1, -1]$, and the possible ultimate response fault patterns are: $P = [0, -1, 0, -1, -1]$, $P = [-1, -1, 0, -1, -1]$, $P = [0, -1, +1, -1, -1]$, and $P = [-1, -1, +1, -1, -1]$.

4.2.2 Fault Caused by Mix Inlet Temperature T_i

The corresponding SDG model for a disturbance caused by inlet temperature T_i is reduced as shown in Figure 4-6. The nodes that cannot be accessed from T_i will stay at their normal state, so measured variables $V = 0$ and $F_o = 0$ are omitted.

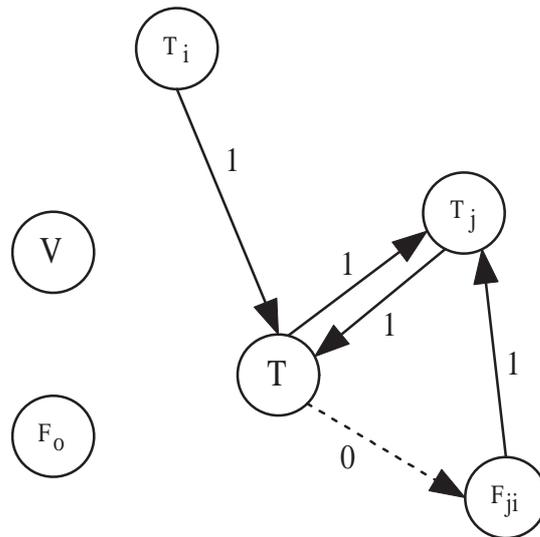


Figure 4-6: SDG Model for Fault T_i

4.2.2.1 Fault Patterns for High Inlet Temperature $T_i(+)$

For disturbance $T_i(+)$, the initial response of T , T_j , and F_{ji} are propagated through the shortest paths as following:

- $T_i \longrightarrow T$, makes $T = +1$,
- $T_i \longrightarrow T \longrightarrow F_{ji}$, makes $F_{ji} = -1$, and
- $T_i \longrightarrow T \longrightarrow F_{ji} \longrightarrow T_j$, makes $T_j = -1$ (by simulation, the other shortest path $T_i \longrightarrow T \longrightarrow T_j$ is not dominant) .

The simulation result for T_j response to T_i high is shown in Figure 4-7, and the process analysis is similar to that for a fault in F_i . One can also use process knowledge (common sense) to reason that an increase in T_i will, by action of the temperature control loop, cause T_j to decrease.

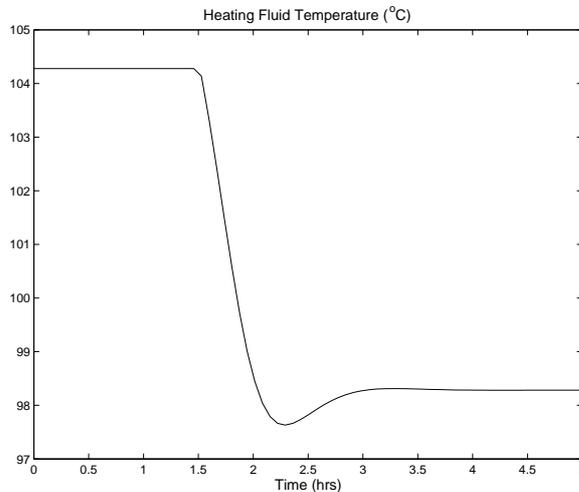


Figure 4-7: T_j Response when T_i High, $t_{0fault} = 1.5$

So the initial response fault pattern of $T_i(+)$ is $P = [0, 0, +1, -1, -1]$.

There only one CV in this SDG, T , so the possible patterns for ultimate response are: $P = [0, 0, 0, -1, -1]$, and $P = [0, 0, +1, -1, -1]$

4.2.2.2 Fault Patterns for Low Inlet Temperature $T_i(-)$

For fault origin $T_i(-)$, the initial response fault pattern is $P = [0, 0, -1, +1, +1]$, the possible patterns for ultimate response are: $P = [0, 0, 0, +1, +1]$, and $P = [0, 0, -1, +1, +1]$.

4.2.3 Fault Caused by Heating Inlet Temperature T_{ji}

The corresponding SDG model for a disturbance in T_{ji} is reduced as shown in Figure 4-8. The nodes that cannot be accessed from T_{ji} , measured variables $V = 0$ and $F_o = 0$, will stay in their normal state.

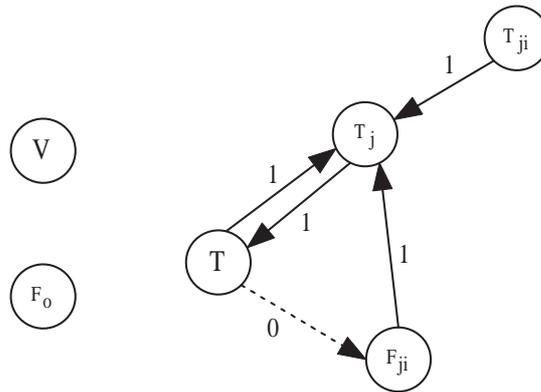


Figure 4-8: SDG Model for Fault T_{ji}

4.2.3.1 Fault Patterns for High Heating Inlet Temperature $T_{ji}(+)$

For fault $T_{ji}(+)$, the initial response of T , T_j , and F_{ji} are propagated through the shortest paths as shown below:

- $T_{ji} \longrightarrow T_j \longrightarrow T$, makes $T = +1$,
- $T_{ji} \longrightarrow T_j$, makes $T_j = +1$, and

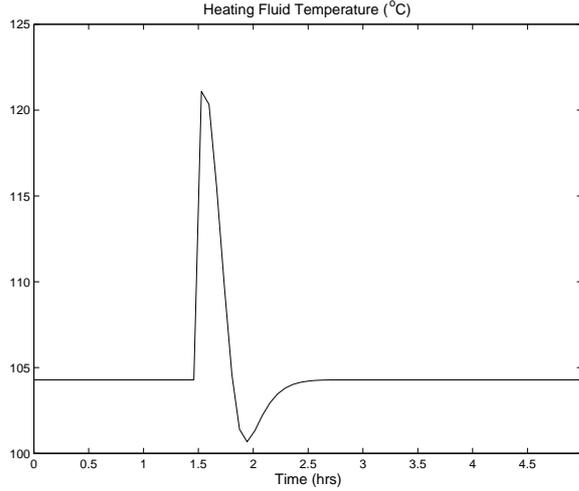


Figure 4-9: T_j Response when T_{ji} High, $t_{0fault} = 1.5$

- $T_{ji} \longrightarrow T_j \longrightarrow T \longrightarrow F_{ji}$, makes $F_{ji} = -1$.

So the initial response fault pattern of $T_{ji}(+)$ is $P = [0, 0, +1, +1, -1]$.

There are two potential CVs in this SDG with respect to disturbance T_{ji} , T and T_j . The variable T_j meets the necessary conditions proposed in [10], see appendix B. First it is located in a negative feedback loop (1). There is only one acyclic path from disturbance T_{ji} to T_j , subsystem $(T_{ji} \longrightarrow T_j)$, and the complementary subsystems are (T, F_{ji}) and \emptyset (the null system). Subsystem (T, F_{ji}) contains one integrator (4a), and none of these subsystems contains a nonzero cycle (4b, 5). So, T_j is a possible CV.

Again, this can be understood by qualitative physics. Because the temperature control loop (TC) is implemented by adjusting T_j (via the heating inflow valve) to control T , if T returns to normal, T_j has to go back to normal. That means either T and T_j are both CVs or neither are CVs. The simulation result of T_j corresponding to a disturbance T_{ji} is shown in Figure 4-9, and illustrates the analysis.

So the possible patterns for ultimate response are: $P = [0, 0, 0, 0, -1]$, and $P = [0, 0, +1, +1, -1]$.

For this process $P = [0, 0, 0, +1, -1]$ will not happen when $T_{ji}(+)$. In addition, from Figure 4-9, it also can predicted that two possible transient response patterns: $P = [0, 0, +1, -1, -1]$ and $P = [0, 0, 0, -1, -1]$.

4.2.3.2 Fault Patterns for Low Heating Fluid Temperature $T_{ji}(-)$

For fault origin $T_{ji}(-)$, the initial response fault pattern is $P = [0, 0, -1, -1, +1]$, the possible patterns for ultimate response are: $P = [0, 0, 0, 0, +1]$, and $P = [0, 0, -1, -1, +1]$.

4.2.4 Volume Sensor Fault, V_s bias

The term sensor fault in this thesis refers to the existence of a sensor bias. The sensor value will be proportional of the true value in some ratio not equal to unity. For example, $V_s = 0.8V$, means that the sensor fault size is -20%. The fault in the simulation model is denoted as $V_s(-)$; to keep the example short the fault $V_s(+)$ is not considered. The reduced SDG model for V_s fault is given in Figure 4-10.

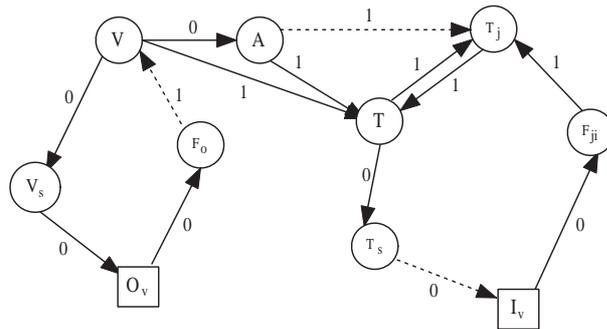


Figure 4-10: SDG Model for V_s Fault

4.2.4.1 Initial Response Fault Pattern for $V_s(-)$

When a negative volume sensor bias occurs, the volume control loop takes action and opens the valve O_v to increase the volume until the sensor measurement shows a return to the normal value. In this case, the SMCC (V_s, O_v, F_o, V) is considered as the fault origin, which is used for propagation outside the control loop. Inside the control loop, the variable deviation is predicted by the shortest path from V_s .

- $V_s = -1$ is used in the patterns, since that is the measured value, even though $V = +1$ is propagated from the path $V_s \longrightarrow F_o \longrightarrow V$,
- $V_s \longrightarrow F_o$, makes $F_o = -1$,
- $V_s \longrightarrow F_o \longrightarrow V \longrightarrow A \longrightarrow T$, makes $T = +1$; $V_s \longrightarrow F_o \longrightarrow V \longrightarrow T$, also makes $T = +1$
- $V_s \longrightarrow F_o \longrightarrow V \longrightarrow A \longrightarrow T_j$, makes $T_j = -1$, and
- $V_s \longrightarrow F_o \longrightarrow V(\longrightarrow A) \longrightarrow T \longrightarrow F_{ji}$, makes $F_{ji} = -1$.

So, the initial response fault pattern is thus $P = [-1, -1, +1, -1, -1]$.

4.2.4.2 Ultimate Response Fault Pattern for $V_s(-)$

In the sensor bias case $V_s(-)$, the controlled variable V cannot return to zero, however, the control loop will stop action when the sensor value returns to its normal value. T would return to the normal state if the temperature control loop works perfectly. The possible final response patterns are: $P = [0, -1, 0, -1, -1]$, $P = [-1, -1, 0, -1, -1]$, and $P = [-1, -1, +1, -1, -1]$. For this process $P = [0, -1, +1, -1, -1]$ will not happen when $V_s(-)$, based on the understanding of the process.

4.2.5 Mix Outflow Valve Fault, O_v Stuck

The valve fault is defined to be a stuck valve, so the flow rate through the valve is fixed. For example, the valve fault size is -20% means that the the outflow rate $F_o = 0.8 * F_{o,normal}$. The fault set in the simulation model is $O_v(-)$. The reduced SDG model for O_v fault is given in Figure 4-11.

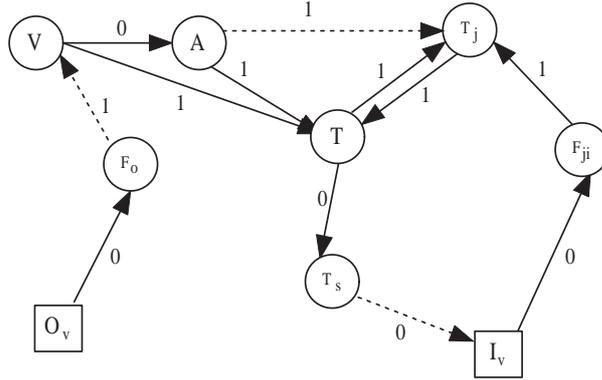


Figure 4-11: SDG Model for O_v Fault

4.2.5.1 Initial Response Fault Pattern for $O_v(-)$

The shortest propagation paths yield the following initial response fault pattern:

- $O_v \rightarrow F_o \rightarrow V$, makes $V = +1$,
- $O_v \rightarrow F_o$, makes $F_o = -1$,
- $O_v \rightarrow F_o \rightarrow V \rightarrow A \rightarrow T$, and $O_v \rightarrow F_o \rightarrow V \rightarrow T$, make $T = +1$,
- $O_v \rightarrow F_o \rightarrow V \rightarrow A \rightarrow T_j$, makes $T_j = -1$, and
- $O_v \rightarrow F_o \rightarrow V(\rightarrow A) \rightarrow T \rightarrow F_{ji}$, makes $F_{ji} = -1$.

The initial response fault pattern thus is $P = [+1, -1, +1, -1, -1]$.

4.2.5.2 Ultimate Response Fault Pattern for $O_v(-)$

In the mix outflow valve fault O_v , the controlled variable V cannot return to zero. So, only T can possibly be compensated by the temperature control loop to a zero deviation state. The possible ultimate response patterns are thus: $P = [+1, -1, 0, -1, -1]$ and $P = [+1, -1, +1, -1, -1]$.

4.2.6 Temperature Sensor Fault, T_s Bias

Again, a sensor fault in this thesis means existence of a sensor bias. For example, $T_s = 0.8T$ means the sensor fault size is -20%. The fault in the simulation model is set as $T_s(-)$. The reduced SDG model for T_s fault is given in Figure 4-12.

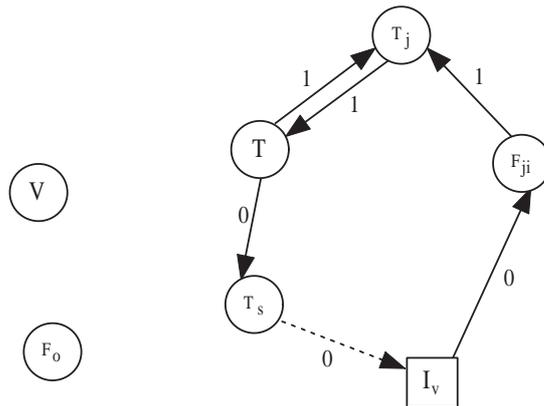


Figure 4-12: SDG Model for T_s Fault

4.2.6.1 Initial Response Fault Pattern for $T_s(-)$

The reduced SDG is an MSCC, but it is the temperature control loop, so variable deviation still can be predicted from the shortest paths. The shortest propagation paths yield the following initial response fault pattern:

- The variables that cannot be accessed from T_s are given deviation values of zero, so $V = 0$ and $F_o = 0$.
- $T_s = -1$ is used in the patterns, although $T = +1$, by propagation from path $T_s \longrightarrow F_{ji} \longrightarrow T_j \longrightarrow T$,
- $T_s \longrightarrow F_{ji}$, makes $F_{ji} = +1$, and
- $T_s \longrightarrow F_{ji} \longrightarrow T_j$, makes $T_j = +1$.

The initial response fault pattern is thus $P = [0, 0, -1, +1, +1]$.

4.2.6.2 Ultimate Response Fault Pattern for $T_s(-)$

In the sensor bias case T_s , the controlled variable T cannot return to zero, however, the control loop will stop taking action when the sensor value return to normal. So, the final possible response pattern are: $P = [0, 0, -1, +1, +1]$ (if the temperature control loop saturates) and $P = [0, 0, 0, +1, +1]$ (if the temperature control loop is effective).

4.2.7 Heating Inflow Valve Fault, I_v stuck

The valve fault is defined to be a stuck valve, so the flow rate through the valve is fixed. For example, the valve fault size is -20% means that the outflow rate $F_{ji} = 0.8 * F_{ji,normal}$. The fault set in the simulation model is $I_v(-)$. The reduced SDG model for I_v fault is given in Figure 4-13.

4.2.7.1 Initial Response Fault Pattern for $I_v(-)$

The shortest propagation paths yield the following initial response fault pattern:

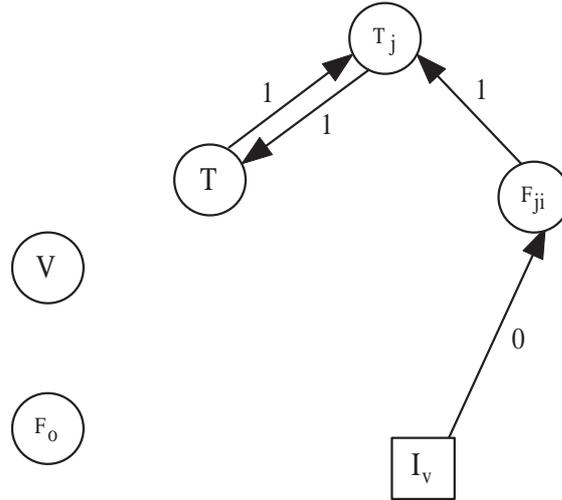


Figure 4-13: SDG Model for I_v Fault

- The variables that cannot be accessed from I_v are given values of zero, so $V = 0$ and $F_o = 0$,
- $I_v \longrightarrow F_{ji}$, makes $F_o = -1$,
- $I_v \longrightarrow F_{ji} \longrightarrow T_j$, makes $T_j = -1$, and
- $I_v \longrightarrow F_{ji} \longrightarrow T_j \longrightarrow T$, makes $T = -1$

The initial response fault pattern is thus $P = [0, 0, -1, -1, -1]$.

4.2.7.2 Ultimate Response Fault Pattern for $I_v(-)$

In the valve fault I_v , the controlled variable T cannot return to normal. So, the final possible response pattern is the same as the initial response: $P = [0, 0, -1, -1, -1]$.

4.2.8 Summary of Fault Patterns

The fault patterns predicted above are possible patterns that might be exhibited by the JCSTR model. Only some of them will appear at one time, according to the

fault type, fault size and compensation capability. All of the prediction patterns were tested by simulating the JCSTR with different fault sizes, the results show the predictions are right. The patterns for each fault are compiled into the IF-THEN rule base, which is used to diagnose the fault in real time. As summary, a pattern table is composed for all the faults by propagation through the SDG, as shown in table 4.5.

	$F_i(-)$	$F_i(+)$	$T_i(-)$	$T_i(+)$	$T_{ji}(-)$	$T_{ji}(+)$	$V_s(-)$	$T_s(-)$	$O_v(-)$	$I_v(-)$
	7	7	6	6	5	5	4	3	2	1
V_s	- 0	+ 0	0	0	0	0	- 0	0	+	0
F_o	-	+	0	0	0	0	-	0	-	0
T_s	+ 0	- 0	- 0	+ 0	- 0	+ 0	+ 0	- 0	+0	-
T_j	-	+	+	-	- 0	+ 0	-	+	-	-
F_{ji}	-	+	+	-	+	-	-	+	-	-

Table 4.5: Possible Patterns Table for JCSTR Model

The first row in the table shows ten faults, including disturbances, valve and sensor faults. The first column lists the five system variables whose qualitative states compose the patterns. For possible compensatory variables, which may return to zero in the final state due to controller action, both qualitative states have been listed, the first sign change is the initial response or possible ultimate response for controller saturation, and the second sign change is zero for perfect control. The number 1 to 7 is the fault representation number or index which will be shown in the fault diagnostic result.

4.3 Simulation Results - Purely Qualitative FDI

The diagnostic system, composed of the IF-THEN rule base which was built by the propagation patterns above, has been tested by introducing one of these ten faults in a simulation, for different fault sizes. The thresholds for the five variables used in

this study are set as 5% of normal values.

The simulation results for a fault size (\pm)20% are shown here. In fact, the diagnosis results are the same for other fault sizes, such as fault size (\pm)50%. See Figure 4-14, Figure 4-15, Figure 4-16, Figure 4-17, and Figure 4-18. For example, in Figure 4.14(a), the x axis is time, and the y axis is the fault diagnosis result. Level 0 is for normal, and levels 1 to 7 stand for the different faults as shown in the pattern table and in the notation F_i , T_i etc. on the y axis. The symbol ‘o’ stands for normal, ‘+’ and ‘ Δ ’ stands for high in initial and final state, ‘x’ and ‘ ∇ ’ stands for low in initial and final state. Two symbols are used here to show the difference between initial and final patterns during a fault for understanding the algorithm and the JCSTR process. The fault is introduced at 1.5 hours. The points corresponding to level 0 mean the system is in a normal operating state. The points corresponding to levels 1 to 7 represent the fault as diagnosed. The result now is good, in the sense that the fault can be isolated in the very beginning, and in the worst cases the list of possible faults is limited to two.

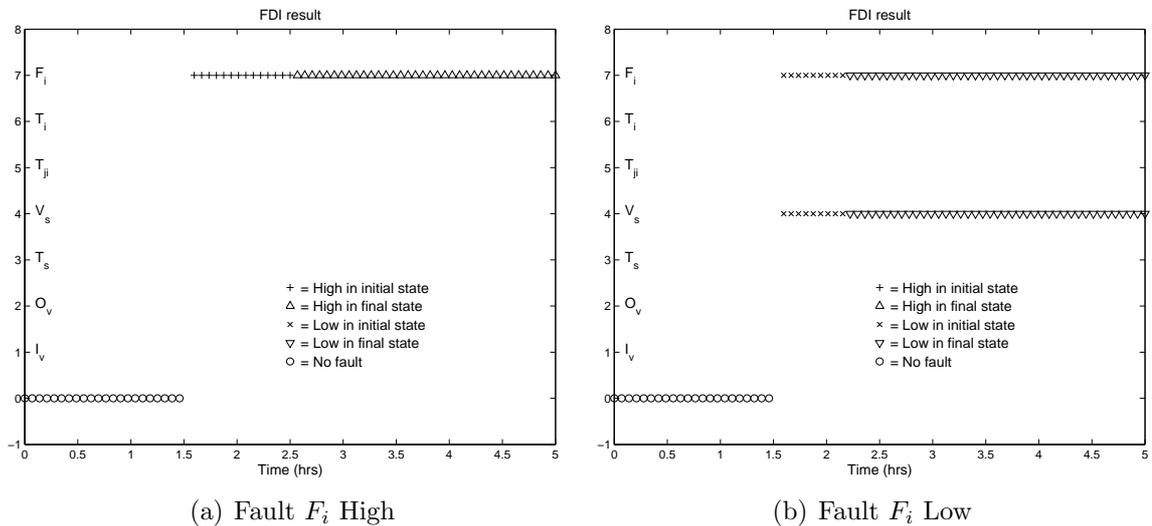
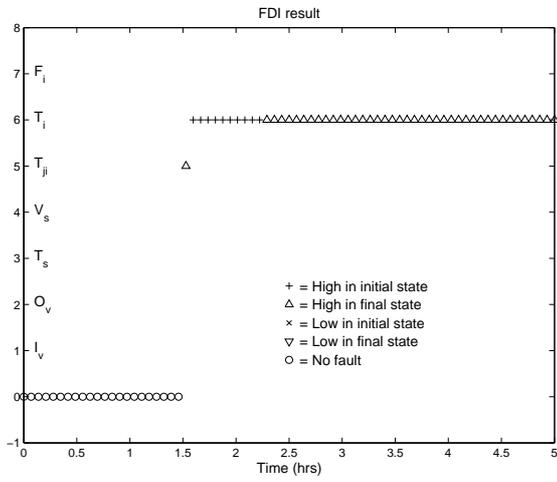
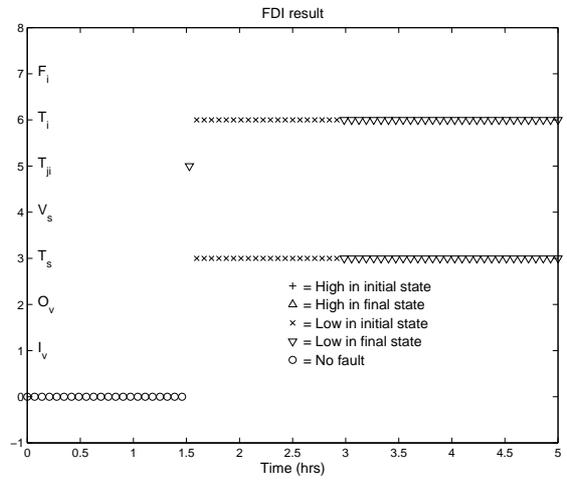


Figure 4-14: FDI Results for Fault F_i

When the abnormal variables are in the neighborhood of the designed threshold,

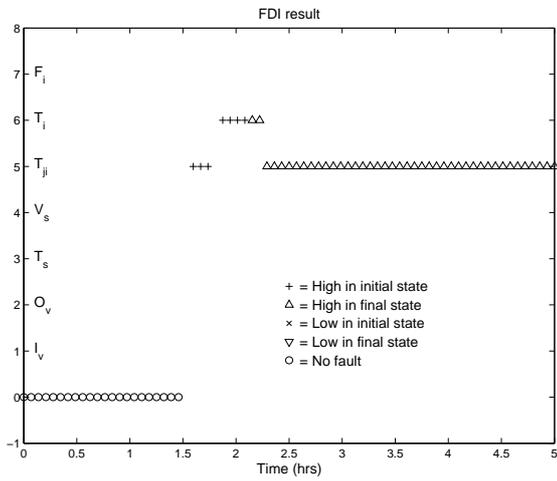


(a) Fault T_i High

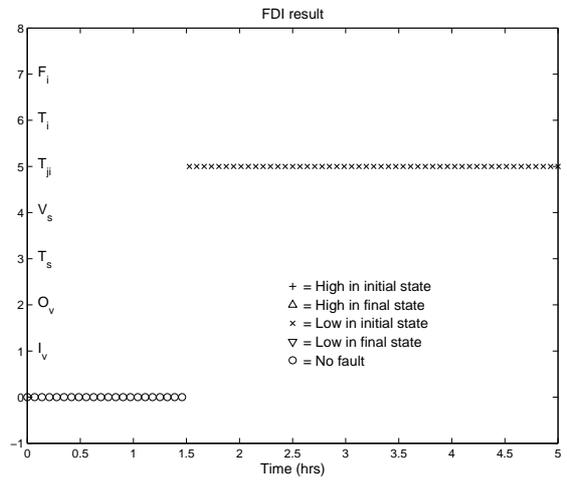


(b) Fault T_i Low

Figure 4-15: FDI Results for Fault T_i

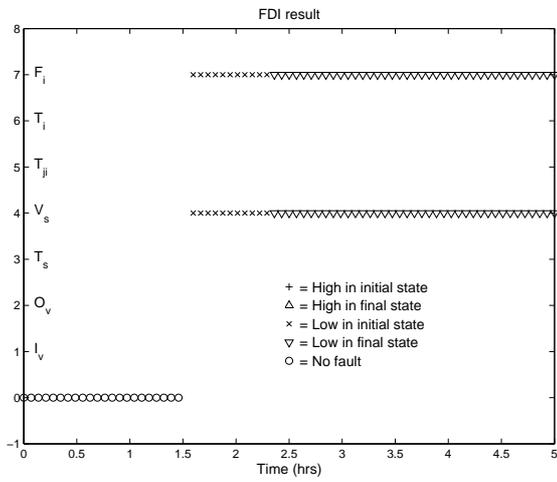


(a) Fault T_{ji} High

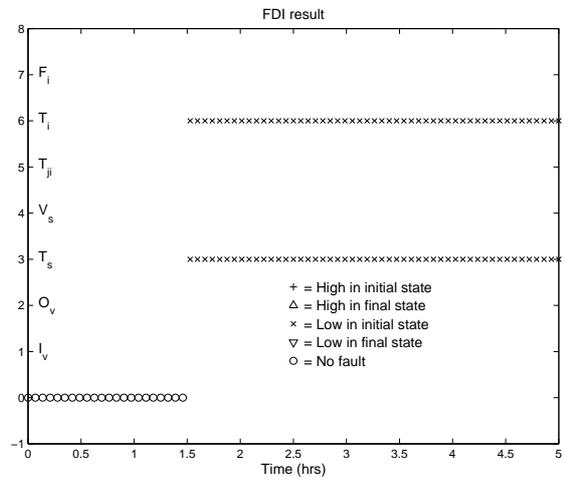


(b) Fault T_{ji} Low

Figure 4-16: FDI Results for Fault T_{ji}

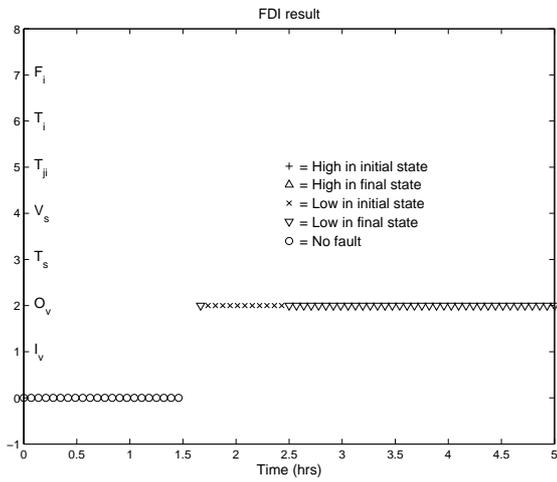


(a) $V_s(-)$ Fault

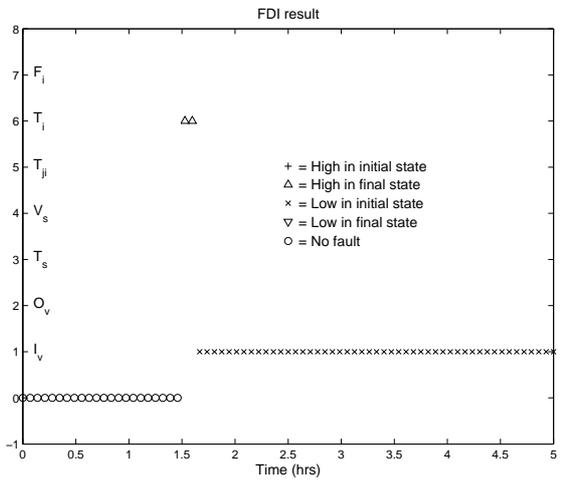


(b) $T_s(-)$ Fault

Figure 4-17: FDI Results for Sensor Faults



(a) $O_v(-)$ Fault



(b) $I_v(-)$ Fault

Figure 4-18: FDI Results for Valve Faults

SDGs have difficulty providing accurate resolution. It can be seen in the figures below. In some of the figures, just after the fault happened and before it has been isolated, there are separate points corresponding to final fault patterns showing up, this is because some abnormal variables' deviation is not large enough to be detected, so its quantitative value is still zero, and unfortunately that matches other patterns. This can be solved by adjusting the thresholds, for example, making them smaller.

The pattern table 4.5 shows that $F_i(-)$ has the same patterns as the $V_s(-)$ sensor fault, and $T_i(-)$ has the same patterns as the $T_s(-)$ sensor fault, so the FDI results also show those faults with same patterns displayed at same time, see Figure 4.14(b) and Figure 4.17(a) and Figure 4.15(b) and Figure 4.17(b). In those figures, the diagnosis result has been limited to the two faults as mentioned; however, these pairs also can be isolated from each other, if one more piece of process knowledge is used, as demonstrated in the next section.

4.4 Hybrid FDI Simulation Results

The sensor faults set in this thesis are step sensor biases, which means that the changes in V_s and T_s produce instantaneous changes in sensed V or T , but changes in F_i or T_i do not, since V and T take time to respond to disturbance inputs based on the underlying dynamics. We take $V_s(-)$ and F_i low as an example, and inspect the different behavior of V as shown in Figure 4-19:

Suppose the fault happens at time sample k , the responses for points $k + 1$ and $k + 2$ are different for these two faults. For the sensor fault, there is an instantaneous change, so $|V(k + 1)|$ is greater than $|V(k + 2)|$, but for the disturbance, $|V(k + 1)|$ is less than $|V(k + 2)|$. Thus only using this two points we can quickly isolate these two faults.

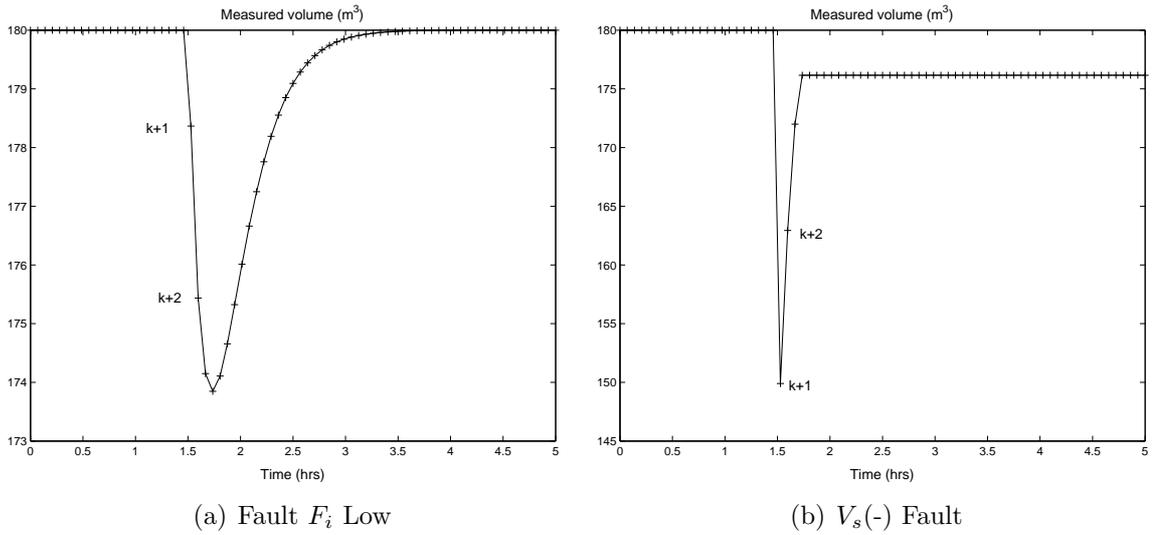


Figure 4-19: Different Behavior of V for Fault F_i low and $V_s(-)$

The diagnostic simulation results using quantitative information are shown in Figure 4-20 and Figure 4-21. The fault size is -20% for each fault. In this scenario, faults with same patterns are correctly isolated. In on-line operation, sampling point $(k+2)$ is unknown at the time of point $(k+1)$, so at time of $k+1$, it still shows another fault if a pattern is matched, as shown in Figure 4.21(b).

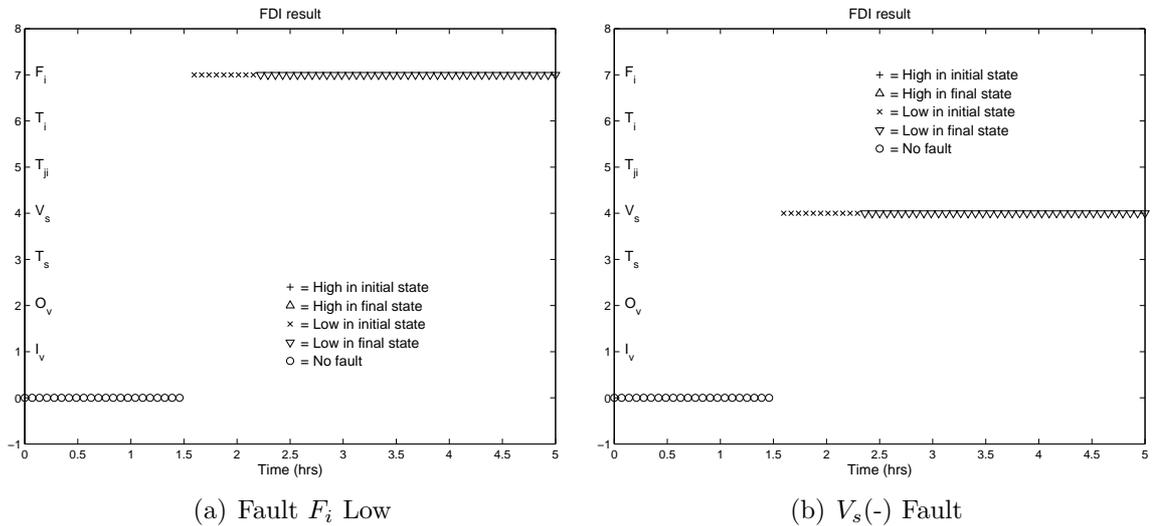


Figure 4-20: FDI Results with Hybrid FDD

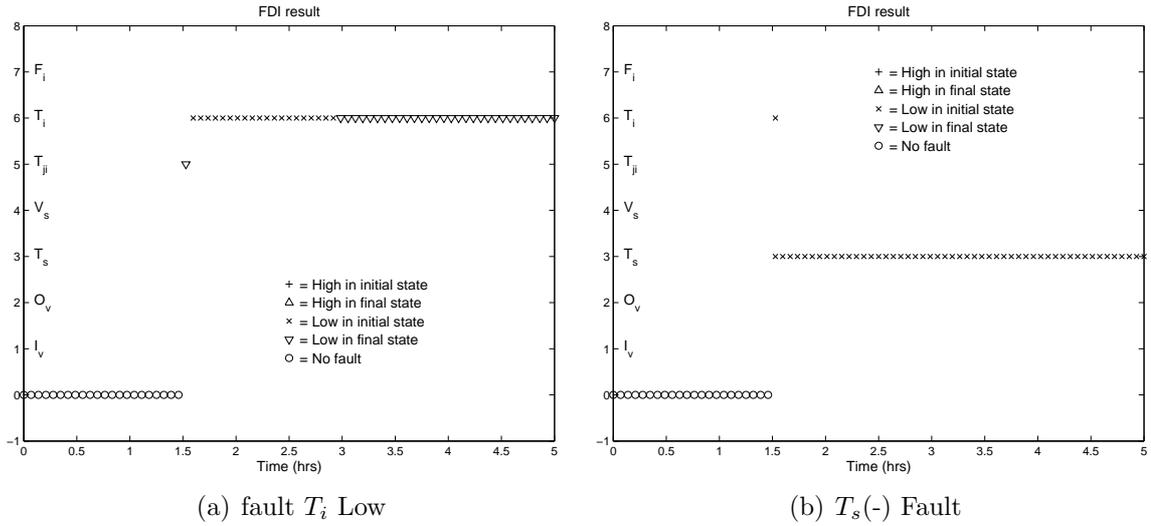


Figure 4-21: FDI Results with Hybrid FDD

4.5 Conclusion

In this chapter, the proposed FDD algorithm has been implemented and applied to a JCSTR model, with a detailed explanation of the procedure and results for all ten potential faults. The most important part of the algorithm development is the propagation of the initial response fault patterns, which determines the operation of the whole diagnostic system, and also affects the early fault diagnosis and isolation. Process knowledge or simulation results were used here to clarify certain ambiguous propagations. The ultimate response fault pattern is determined by the correct identification of CVs and IVs, and similar decisions have to be made for the analysis of any real system. Quantitative data has been used to distinguish the faults with same qualitative patterns to further increase the diagnostic resolution.

The simulation results for the JCSTR model show that the overall algorithm is efficient and promising. All ten potential faults can be isolated correctly and in a very short time, fulfilling the requirement for early and correct fault diagnosis and isolation.

Chapter 5

Robustness of the Fault Diagnostic System

The simulation results shown in the last chapter are based on ideal conditions, and faults are introduced at times when the system is in steady state operation. In reality, the operation is uncertain, and faults can happen at anytime, thus the problem of robustness of the diagnostic algorithm arises. This issue is addressed in this chapter.

Robust fault diagnosis refers to the ability of fault diagnostic systems to make a correct diagnosis in the presence of operation uncertainties and various perturbations. A robust diagnostic system's performance should degrade gracefully instead of failing totally and abruptly.

In an SDG-based approach, three qualitative states, normal (0), high (+), and low (-), are used to present a variable which is equal to, above, or below the steady state value, respectively. In diagnosing the process, thresholds are used to calculate the qualitative states. In the ideal conditions, thresholds are set close to zero, which benefits sensitivity. The threshold sets the tolerance of a variable to noise and fluctuation.

However, in setting the tolerance, compromises have to be made between detection sensitivity and unnecessary alarms due to normal fluctuations. So thresholds and steady state values are the factors that affect the result of qualitative analysis, and they should come into consideration when regarding robustness.

In this chapter four aspects will be addressed regarding the robustness problem: (1) different thresholds; (2) dealing with transients; (3) operation point variation; and (4) operation fluctuation. In simulations for cases (2), (3) and (4), we suppose that another fault detection approach is used, due to the limitation of limit value checking for fault detection, so in those cases we suppose that we know the time that the fault happened.

5.1 Different Thresholds

Table 5.1 gives two sets of thresholds for the five variables used in the JCSTR model. The first set is the one used in last chapter, and the second set is much looser than the first one and it is selected by guaranteeing that all the ten faults can be isolated correctly in normal conditions. The JCSTR FDI system was designed using an ideal model with no noise and no fluctuation, so the thresholds were previously set small.

Variables	Threshold (1)	Threshold (2)
V	$0.5\%V_{normal}$	$3\%V_{normal}$
F_o	$0.5\%F_{o,normal}$	$3\%F_{o,normal}$
T	$0.5\%T_{normal}$	$2\%T_{normal}$
T_j	$0.5\%T_{j,normal}$	$3\%T_{j,normal}$
F_{ji}	$0.5\%F_{ji,normal}$	$3\%F_{ji,normal}$

Table 5.1: Comparison Two Sets of Thresholds

The algorithm has been tested using both sets of thresholds. The disturbance $F_i(+)$ (high mix inflow) is used as an example fault, and it is introduced at 1.5 hours in the

steady state operation period. The FDI results corresponding to two fault sizes 10% and 20% are shown here, see Figures 5-1 and 5-2.

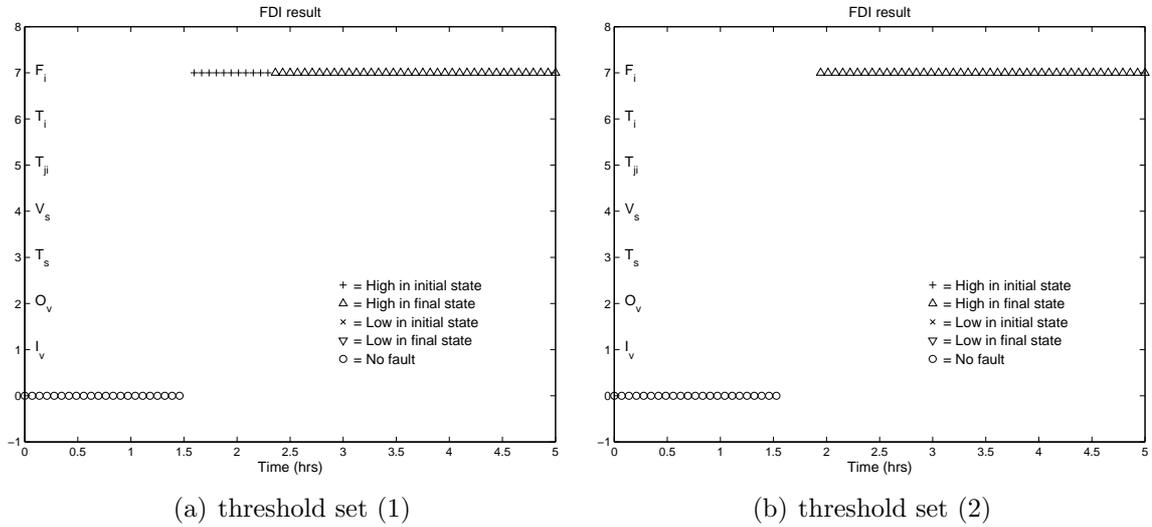


Figure 5-1: FDI Result, Different Thresholds for Fault Size 10%

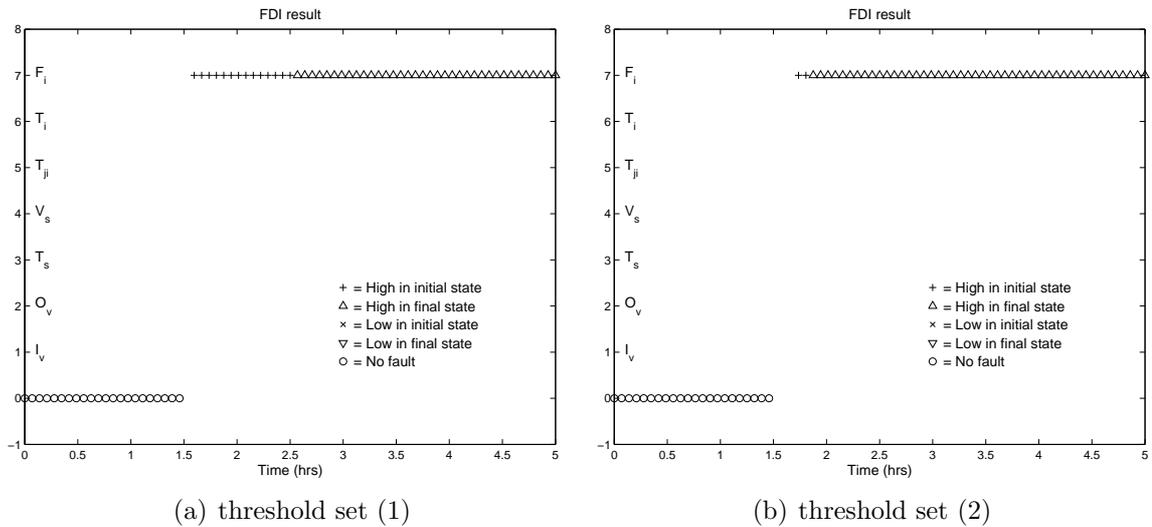


Figure 5-2: FDI Result, Different Thresholds for Fault Size 20%

For both fault sizes, the fault is correctly isolated, although, for threshold set (1), which is smaller, the time for isolation is earlier than for threshold set (2), which is bigger. This demonstrates that increasing thresholds decreases the detection sensitivity for FDI. This slow behavior is reduced if the fault size increases, however.

For threshold set (2), the time of isolation for a 20% fault is earlier than for the smaller 10% fault. It can be deduced that larger thresholds will affect fault detection if the fault falls into the neighborhood of threshold, and isolation time is much more affected for smaller faults.

Moreover, the sampling points which matched the initial response and ultimate response patterns are different. For the smaller threshold set or bigger fault size, more points corresponding to initial patterns are identified; for the bigger threshold set or smaller fault size, fewer points correspond to initial patterns; there are no initial pattern points in the case shown in Figure 5.1(b).

If the fault size is large enough to provide changes that exceed the thresholds, then the FDI result is still correct; however, it raises the issue that the diagnostic algorithm might have problems when the fault is small, which is the object of future research. So although sometimes we need to increase thresholds, for example, if system noise exists, compromises have to be made between the detection sensitivity (both fault size and isolation time) and unnecessary alarms.

5.2 Dealing with Transients

The SDG-based approach uses steady state values of measured variables to calculate their qualitative states, which compose patterns used in fault diagnosis. So a diagnostic system needs to know the steady state value of the measured variables at different operation points, for example, the steady state for tank volume at 100% capacity is different with the one for volume at 80%. When there is a need to change the operation point, the corresponding steady state values need also to be supplied as an input to the diagnostic system for correct diagnostic results, as shown in Figure 3-1. We assume there is a look-up table corresponding to different operation points

based on the changes of temperature and volume set points. Once the operation point changes, the steady state values at the new operation point are used as nominal values for FDI.

There is a transient period during operation changes. Since the steady state values of the new operation point are used as nominal value for FDI, the variables' qualitative states may not be zero during the transient period, since the variables have not arrived at steady state yet. However, since the diagnostic system keeps calculating the patterns, even if there are no faults during the transient period there may be alarms and faults declared, as shown in Figure 5-3, in which the operating point is changed by increasing temperature setpoint by 5% and the volume setpoint by 10% at time 0 hour, and it arrives at a new steady state at around 2 hours.

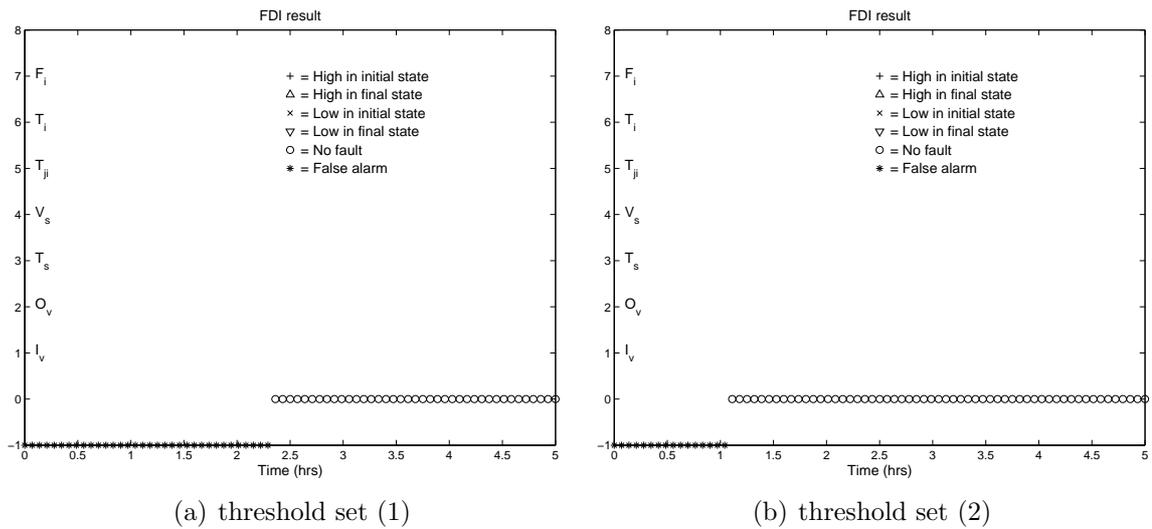


Figure 5-3: FDI Result, Different Thresholds for Normal Operation

The system is in a fault free condition, however, alarms or false faults are identified during the transient period because variables have not arrived at new nominal values. For the time intervals where “*” is displayed at level -1, this indicates that some variables are not in normal conditions, so an alarm is raised. In this thesis, fault patterns in the pattern set are not compared during transient periods, so any patterns

that do not match with the normal pattern, $P = [0, 0, 0, 0, 0]$, are shown as false alarm. This is based on the assumption that we know that this interval is during an operation point change.

An important difference is shown in these two figures before time 2.5 hours. The difference in thresholds affects the displayed time when the system arrives at the new steady state. When the variables' quantitative values are in the neighborhood of their thresholds, different threshold settings will produce different qualitative value, and this affects the indication of steady state points (corresponding to value 0) shown in the figures.

We normally suppose, as shown in the last chapter, that faults happen in periods of steady operation, however, faults can happen during the transient period. If the new steady state nominal value is used, the FDI algorithm still can isolate the ten faults correctly, although the performance will not be as perfect. Here we use high mix inflow as an example, see Figure 5-4; all other simulation results are given in Appendix C.1, Figures C-1 to C-5. The faults are introduced during steady state at

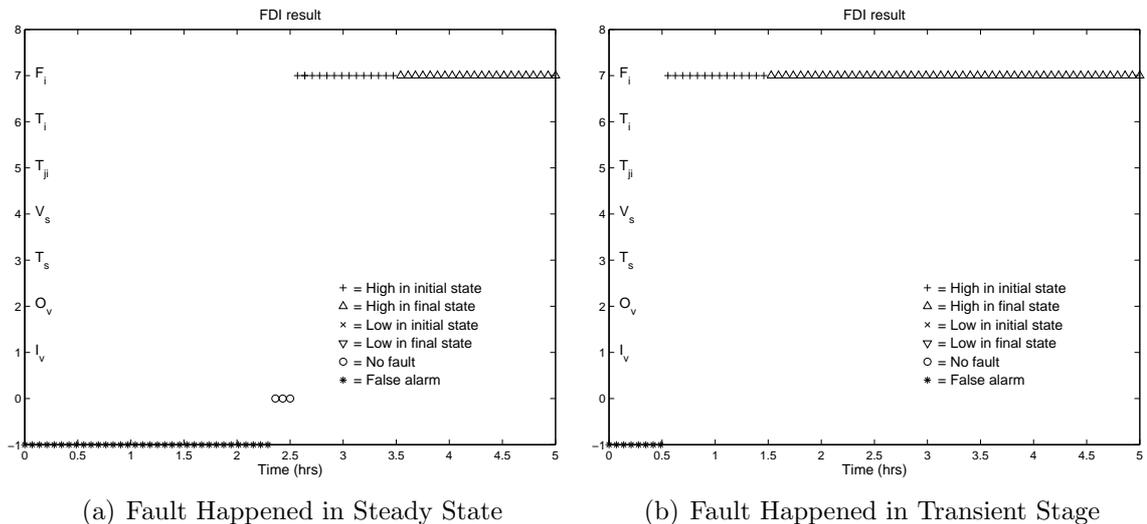


Figure 5-4: FDI Result, Fault Happened in Transient Period

time 2.5 hours and during the transient period at time 0.5 hours, respectively, with

fault size 20% and threshold set (1). As mentioned, we require another detection method since the SDG algorithm only detects faults by watching for the pattern to change from $P = [0, 0, 0, 0, 0]$.

5.3 Operation Point Variation

Although the nominal values for the algorithm have to be set accurately, it is not highly critical; the FDI algorithm with one set of nominal values works over a small region of operation point variation.

Figure 5-5 shows the workable region around the nominal operation point, i.e., $dV_{sp} = 0$ and $dT_{sp} = 0$. The meanings of the symbols in the figure are: “o”, FDI result is

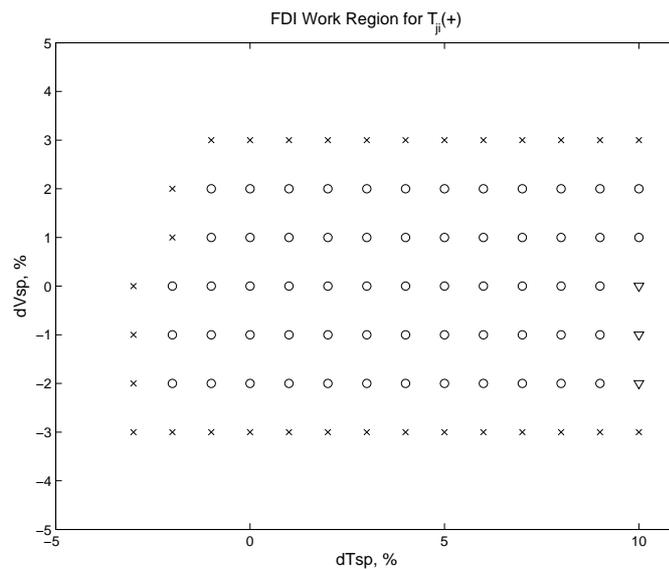


Figure 5-5: FDI Work Region

correct and fast, less than one hour; “∇”, FDI result is correct but late, more than one hour and less than two hours; “x”, FDI was not possible, or very late, or only several points (less than 3) are shown. Two sample plots for understanding the symbols also shown in Figure 5-6. These figures show the affect of high heating fluid temperature,

fault size 20% and/or 10%, as an example, and the threshold set in this example is threshold set (2) in Table 5.1. Compared with Figure 4.16(a), the results are quite acceptable. Figure 5.6(a) corresponds to point $dT_{sp} = 1\%$, $dV_{sp} = 1\%$ and Figure 5.6(b) corresponds to point $dT_{sp} = 10\%$, $dV_{sp} = -2\%$. It can be seen that there is

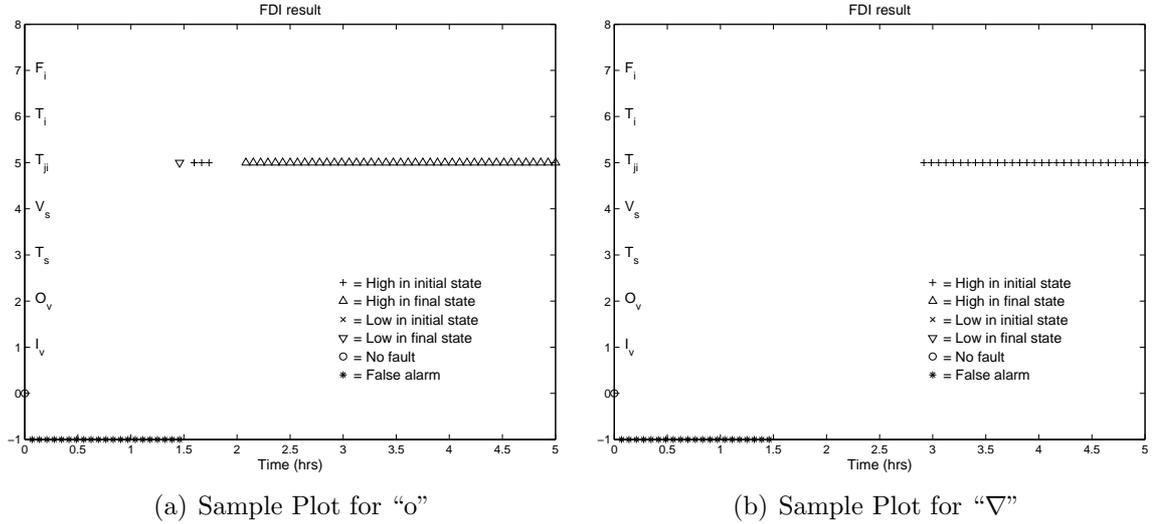


Figure 5-6: Example FDI Results

a small area where the FDI system works for the nominal values. For other faults, although the region areas with “∇” are different, they all have a small working region. See Appendix C.2 for detail.

If the threshold setting is much bigger, the working region will be little bigger. But, the threshold setting should be small enough to ensure that it works for the ten faults in normal conditions.

5.4 Operation Fluctuation

The simulation results shown in the last chapter are based on the system without any perturbation, and the operating condition is nominal. In reality, process operation is accompanied by normal fluctuations and the system response is not as smooth as

shown in simulation. The robustness test with operational fluctuation has been done using the simulation model, with a sinusoidal signal added to one controller set point at a time, to simulate slow fluctuations.

The simulated fluctuation should be set realistically. The JCSTR model used in this thesis is a tank with volume of 180 m^3 , and heating jacket with volume of 9m^3 , so it is reasonable to set the tank volume maximum changes at $\pm 5\%$, the maximum temperature changes at $\pm 2\%$, and the change frequency is set low. In the robustness test, two different frequencies were studied, with periods 10 hours and 5 hours, so if the fault appears at 1.5 hours, we simulate two conditions: the fault is introduced at the times of intermediate value and maximum value of the sinusoidal wave. Four cases are studied, as shown in Table 5.2:

Case A	$dT_{sp} = 0.02 \sin(\frac{2\pi}{3600*10}t) * T_{sp}$ $dV_{sp} = 0$
Case B	$dT_{sp} = 0$ $dV_{sp} = 0.05 \sin(\frac{2\pi}{3600*10}t) * V_{sp}$
Case C	$dT_{sp} = 0.02 \sin(\frac{2\pi}{3600*5}t) * T_{sp}$ $dV_{sp} = 0$
Case D	$dT_{sp} = 0$ $dV_{sp} = 0.05 \sin(\frac{2\pi}{3600*5}t) * V_{sp}$

Table 5.2: Four Cases of Operation Fluctuation

The FDI results for the four cases are shown in table 5.3 and table 5.4. In these tables, isolation “yes 1” means the corresponding fault is isolated correctly; “yes 2” means there is a second false fault identified along with the real fault, while “late” means the real fault is diagnosed only after one hour. A “yes” under “False alarm” means that in normal operations (before 1.5 hours), a false fault has been declared, though there was no fault.

Figure 5.7(a) and Figure 5.7(b) are samples of the FDI results and shown here to illustrate the notations in the tables. The FDI results for all the faults in all the

Fault	case A		case B	
	Isolation	False alarm	Isolation	False alarm
$F_i(+)$	yes 1	yes	yes 1	no
$F_i(-)$	yes 1	yes	yes 1	no
$T_i(+)$	yes 2, late	yes	yes 1	no
$T_i(-)$	yes 2	yes	yes 2, late	no
$T_{ji}(+)$	yes 1	yes	yes 1	no
$T_{ji}(-)$	yes 1	yes	yes 1	no
V_s	yes 1, late	yes	yes 1	no
T_s	yes 1	yes	yes 1	no
O_v	yes 1, late	yes	yes 1	no
I_v	yes 2	yes	yes 2, late	no

Table 5.3: FDI Results for Robustness Test, 1

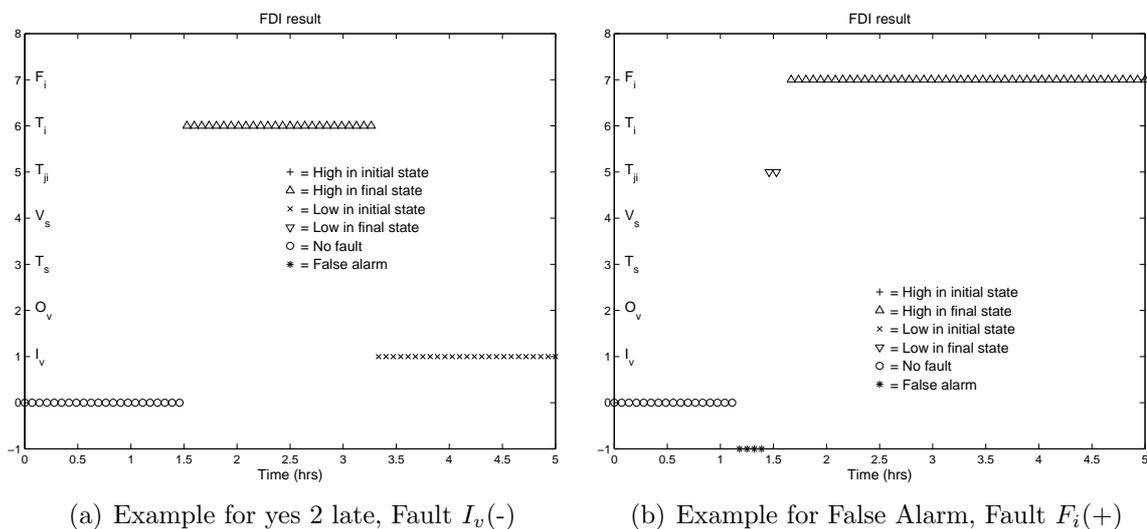


Figure 5-7: Example FDI Results

cases are shown in Appendix C.3. In those simulations, the system is in normal state since time 0, and the fault is introduced at 1.5 hours. All fault sizes are 20%. The thresholds for the five variables used here are listed in Table 5.5.

The threshold used here are much looser than the ones used before, but they are not set unreasonably large enough to cancel all the false alarm during normal condition. If that was done, some faults would not be identifiable. All the false alarms shown

Fault	case C		case D	
	Isolation	False alarm	Isolation	False alarm
$F_i(+)$	yes 1	yes	yes 1	yes
$F_i(-)$	yes 1	yes	yes 1	yes
$T_i(+)$	yes 1	yes	yes 2	yes
$T_i(-)$	yes 2	yes	yes 2, late	yes
$T_{ji}(+)$	yes 1	yes	yes 1	yes
$T_{ji}(-)$	yes 1	yes	yes 1	yes
V_s	yes 1	yes	yes 1	yes
T_s	yes 1	yes	yes 1	yes
O_v	yes 1	yes	yes 1	yes
I_v	yes 2	yes	yes 2	yes

Table 5.4: FDI Results for Robustness Test, 2

Variables	Threshold
V	$6\%V_{normal}$
F_o	$6\%F_{o,normal}$
T	$3\%T_{normal}$
T_j	$4\%T_{j,normal}$
F_{ji}	$16\%F_{ji,normal}$

Table 5.5: Thresholds Used in Robustness Tests

here are caused by large variation in F_{ji} , so if the operator is aware this, or if such process knowledge is used in the FDI system, it would not be a problem.

From the FDI results shown in Appendix C.3, we see that all ten faults can be diagnosed, although in a few of cases a false fault appears with the real fault; except for six fault cases, all other faults can be diagnosed in early time.

5.5 Conclusion

In this chapter four problems related to robustness have been investigated. Based on the discussion and results, it is shown that the algorithm still performs well without

changing any of the rules compiled from the SDG model. If fluctuation is encountered during system operation, the diagnostic algorithm degrades gracefully, instead of suddenly. Also, we showed that helpful tradeoffs can be made in setting thresholds to handle different robustness issues. The proposed diagnostic system is thus robust.

Chapter 6

Conclusions and Suggestion for Future Developments

6.1 Main Conclusions

The new ideas and studies reported in this thesis encapsulate the contribution to creating an enhanced SDG qualitative model based approach for fault detection and diagnosis. The thesis objective is to develop an effective computer-aided on-line fault diagnosis technique in order to keep the system performance as close as possible to the optimal. A systematic design procedure for constructing a rule based fault diagnostic system using SDGs is proposed and implemented. The whole procedure is demonstrated by application to the JCSTR model, and the simulation results show that the algorithm is efficient and robust.

Unlike quantitative approaches, which require a rigorous process model and extensive measurement to collect process data for parameter estimation, qualitative approaches based on signed digraphs (SDGs) can clearly express qualitative relationships between

variables, and only a minimum of process data is required to perform a quick diagnosis. The biggest drawback of the SDG based approach is the loss of diagnosis resolution, because of its qualitative nature.

A large number of articles in the literature focus on different ways of modifying the SDG model to improve the ability of search, however in this thesis efforts have been made firstly on enumerating precisely all possible qualitative states of all the measured variables through propagating the original SDG model, both for initial and ultimate response, and secondly on using process knowledge to good advantage, so that the diagnostic resolution has been increased significantly.

Taking advantage of both initial and ultimate response patterns for fault isolation makes early fault diagnosis possible and isolation more accurate. The concept of initial response has been adopted by other authors; however, because of the precise propagation of the pattern in this research, the advantage of initial response has been exploited as widely as possible.

This research proposed a new way to understand PI control loops in an SDG, and to propagate initial response patterns of control loops that is simple but efficient. Using process knowledge to isolate sensor faults and disturbances of controlled variables has also been done in this thesis for the first time, increasing diagnosis resolution significantly.

New results were also obtained in the identification of compensatory variables (CVs) and inverse variables (IVs). Although the necessary conditions are clearly stated regarding these two kinds of variables, they are not sufficient. Process knowledge and analysis is needed to obtain correct results. If the system has many controllers or IVs and CVs, the work load and the size of the possible pattern base corresponding to each states will seem to be large, however, with additional process knowledge, the number of patterns can be reduced significantly, since generally compensatory

variables arrive at steady state with only certain patterns.

The shortcoming of the algorithm comes from the use of process knowledge. Process information is used during the propagation and building the quantitative rule base; this makes the algorithm much more case specific. For each case we have to develop the corresponding knowledge base based on system study.

The algorithm proposed in this thesis is a good fault diagnosis and isolation approach, however it has limited capabilities for fault detection during transients or fluctuating operation, so combining another fault detection technique with this SDG based FDI approach may make it more effective.

6.2 Suggestions for Future Work

Research of this scale is obviously insufficient to cover every aspect of the topic. This work has left much opportunity for further development.

This algorithm was only thoroughly tested and shown effective in steady state; only a small effort was focussed on transient operation. Future work should pay more attention to transient FDI.

When abnormal variables are in the neighborhood of the designed thresholds, SDGs have difficulty providing accurate resolution. This thesis has not done a comprehensive study of this aspect. However, the diagnostic algorithm proposed here is based on a set of logic rules, which can be combined with other rules pertaining to plant operations in an expert system. In principle, the expert systems approach is amenable to the use of fuzzy logic, which may help address the problem of alarm threshold sensitivity [9].

The effects of time delay and large time constants were not considered in this thesis.

The algorithm proposed here could solve some problems related to time delay and differences in time constant, if they are not extreme; however, when the time constants are very different, the patterns taken from the same time sample may be wrong for use as patterns in the rule base. This could be solved, at least partly, by calculation of the relative time offsets and taking measurements at different sample times to obtain the pattern to be compared. However, this will require further work and be more case specific.

In this thesis, the sequence and time to fire different patterns, initial and ultimate response patterns, was not taken into consideration, this is yet another future area for research. For large systems, one fault's initial response pattern may be the same with other ultimate response pattern, so it is necessary to set up a time sequence, first looking at initial response patterns, later on looking at final patterns.

Further research also should be done on large flow sheet systems. The bigger the system, the more variables, and the more complicated it is to isolate faults correctly. Though the system could still be analyzed, the size and complexity of this job may be enormous.

Bibliography

- [1] D. M. Himmelblau. *Fault detection and diagnosis in chemical and petrochemical processes*. Elsevier press, Amsterdam, 1978.
- [2] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N. Kavuri. A review of process fault detection and diagnosis part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27(3):293–311, 2003.
- [3] Venkat Venkatasubramanian, Raghunathan Rengaswamy, and Surya N. Kavuri. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3):313–326, Mar 2003.
- [4] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri, and Kewen Yin. A review of process fault detection and diagnosis part III: Process history based methods. *Computers and Chemical Engineering*, 27(3):327–346, 2003.
- [5] M. Iri, K. Aoki, E. O’Shima, and H. Matsuyama. An algorithm for diagnosis of system failures in the chemical process. *Computers and Chemical Engineering*, 3:489–493, 1979.
- [6] T. Umeda, T. Kuriyama, E. O’Shima, and H. Matsuyama. A graphical approach to cause and effect analysis of chemical processing systems. *Chemical Engineering Science*, 35(12):2379–2388, 1980.
- [7] M. Kokawa, S. Miyazaki, and S. Shingai. Fault location using digraph and inverse directon search with application. *Automatica*, 19:729–735, 1983.
- [8] J. Shiozaki, H. Matsuyama, K. Tano, and E. OShima. Fault diagnosis of chemical processes by the use of signed, directed graphs: Extension to five-range patterns of abnormality. *International Chemical Engineering*, 25(4):651–659, 1985.

- [9] M. A. Kramer and B. L. Palowitch Jr. Rule-based approach to fault diagnosis using the signed directed graph. *AIChE Journal*, 33(7):1067–1078, Jul 1987.
- [10] Olayiwola Oyeleye and Mark A. Kramer. Qualitative simulation of chemical process systems: Steady-state analysis. *AIChE Journal*, 34(9):1441–1454, Sep 1988.
- [11] Chung-Chien Chang and Cheng-Ching Yu. On-line fault diagnosis using the signed directed graph. *Industrial and Engineering Chemistry Research*, 29(7):1290–1299, Jul 1990.
- [12] N.A. Wilcox and D.M. Himmelblau. Possible cause and effect graphs (PCEG) model for fault diagnosis - I. methodology. *Computers and Chemical Engineering*, 18(2):103–116, Feb 1994.
- [13] N.A. Wilcox and D.M. Himmelblau. Possible cause and effect graphs (PCEG) model for fault diagnosis - II. applications. *Computers and Chemical Engineering*, 18(2):117–127, Feb 1994.
- [14] David Leung and Jose Romagnoli. Dynamic probabilistic model-based expert system for fault diagnosis. *Computers and Chemical Engineering*, 24:2473–2492, 2000.
- [15] Chung-cheng Han, Ruey-fu Shih, and Liang-sun Lee. Quantifying signed direct graphs with the fuzzy set for fault diagnosis resolution improvement. *Industrial and Engineering Chemistry Research*, 33(8):1943–1954, Aug 1994.
- [16] Ruey-Fu Shih and Liang-Sun Lee. Use of fuzzy cause - effect digraph for resolution fault diagnosis for process plants. 1. fuzzy cause - effect digraph. *Industrial and Engineering Chemistry Research*, 34(5):1688–1702, May 1995.
- [17] Ruey-Fu Shih and Liang-Sun Lee. Use of fuzzy cause - effect digraph for resolution fault diagnosis for process plants. 2. diagnostic algorithm and applications. *Industrial and Engineering Chemistry Research*, 34(5):1703–1717, May 1995.
- [18] Enrique E. Tarifa and Nicolas J. Scenna. Fault diagnosis, direct graphs, and fuzzy logic. *Computers and Chemical Engineering*, 21(SUPPL):S649–S654, May 1997.
- [19] Enrique E. Tarifa and Nicolas J. Scenna. Fault diagnosis for a MSF using a SDG and fuzzy logic. *Desalination*, 152(1-3):207–214, Feb 10 2002.

- [20] D. Mylaraswamy, S. Kavuri, and V. Venkatasubramanian. A framework for automated development of causal models for fault diagnosis. *AIChE Annual Meeting, Miami*, page 232g, 1994.
- [21] Mano Ram Maurya, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. A systematic framework for the development and analysis of signed digraphs for chemical processes. 1. algorithms and analysis. *Industrial and Engineering Chemistry Research*, 42(20):4789–4810, Oct 2003.
- [22] Mano Ram Maurya, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. A systematic framework for the development and analysis of signed digraphs for chemical processes. 2. control loops and flowsheet analysis. *Industrial and Engineering Chemistry Research*, 42(20):4811–4827, Oct 2003.
- [23] Mano Ram Maurya, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. Application of signed digraphs-based analysis for fault diagnosis of chemical process flowsheets. *Engineering Applications of Artificial Intelligence*, 17(5):501–518, August 2004.
- [24] Dinkar Mylaraswamy and Venkat Venkatasubramanian. A hybrid framework for large scale process fault diagnosis. *Computers and Chemical Engineering*, 21(SUPPL):S935–S940, May 1997.
- [25] Zhao-qian Zhang, Chong-guang Wu, Bei-ke Zhang, Tao Xia, and An-feng Li. Sdg multiple fault diagnosis by real-time inverse inference. *Reliability Engineering and System Safety*, 87(2):173–189, February 2005.
- [26] Claire Palmer and Paul W. H. Chung. Eliminating ambiguities in qualitative causal feedback. *Computers chem. Engng*, 22:S843–S846, 1998.
- [27] Mylaraswamy Dinkar. *DKit: A blackboard-based, distributed, multi-expert environment for abnormal situation management*. PhD thesis, PURDUE UNIVERSITY, 1996.
- [28] Atalla Sayda. A benchmark model of a jacketed stirred tank heater for fault detection and isolation. Technical report, University of New Brunswick, Fredericton, NB, Canada, May 2004.

Appendix A

SDG Concepts

The following definitions are given for the purpose of understanding graph theory and SDG analysis. They are taken from the literature [5], [9], [10] and [21].

Truth Table:The SDG propagation rule or qualitative manipulation used in this thesis is shown in the following table provided in Figure A-1. In the truth table, the

		A \longrightarrow B			A $\cdots\cdots\longrightarrow$ B		
		B	1	0	-1	1	0
A	1	T	F	F	F	F	T
	0	F	T	F	F	T	F
	-1	F	F	T	T	F	F

Figure A-1: Truth Table for Qualitative Simulation

letter T (TRUE) corresponds to a consistent branch according to the measurement pattern of the initial and terminal nodes. The above truth table does not consider time delay and/or disturbance damping on the branch. For example, suppose that the initial node A take value of '+' or '-', if we consider time delay or time lag, then the terminal node B may deviate from the normal value or remain unchanged. However, under this truth table, all of the variables will have either positive or negative deviation. A normal $A(A = 0)$ or normal $B(B = 0)$ implies fault does not propagate through the branch.

Directed Path: A directed path from node A to node B in a digraph is an alternating

sequence of nodes and directed arcs of the digraph such that the first and last nodes in the sequence are nodes A and B, respectively.

Consistent Branch: the initial node and the terminal node match the sign on the arc according to the qualitative manipulation.

Strongly Connected Component (SCC): A subset of a digraph is called a strongly connected component if every node of the subset can be reached from every other node of this subset.

Maximum Strongly Connected Component (MSCC): A node or SCC with no input arcs. An MSCC is sometimes also called a root node.

The complement subsystem to an acyclic path in the SDG is the subgraph that is obtained if all nodes in the acyclic path (including initial and terminal nodes) are eliminated.

The complement subsystem to a cycle path in a subgraph of the SDG is the subgraph obtained if all nodes in the cycle are eliminated from the original subgraph.

Initial Response: The initial response of a system variables is its first nonzero response. The initial response of the entire system is its response at the smallest time by which all of the system variables have shown their initial response. It is also called *the first change* in some place. There may be more than one possible initial response.

Final or ultimate response: A reachable state where each node with an unambiguous net influence has the same sign as the influence. There may be multiple possible final responses.

Inverse Response (IR): the final sign of a variable is opposite from the initial direction of deviation of the variable.

Compensatory Response (CR): the variable returns to its nominal steady-state value after an initial deviation.

Inverse Variables (IVs) and Compensatory Variables (CVs): are defined as variables that exhibit IR or CR to a particular disturbance due to negative feedback.

Arc length in the SDG of a DAE System are determined as follows: The arcs in the DE part of the SDG have length 1 and the arcs in the AE part of the SDG have arc length 0 because of the instantaneous response behaviors [21].

Shortest path(s): Shortest path(s) from an exogenous variable (e_l) to a system

variable (x_j) is the directed path with the smallest number of arcs length among all of the directed paths from e_i to x_j .

Pattern: If a fault happens, there are deviations in process variables, and the set of all the symptoms caused by a fault is the pattern of this fault.

Appendix B

Necessary Conditions for CVs and IVs

As first proposed by Oyeleye and Kramer [10], generally, (1) inverse variables (IVs) and compensatory variables (CVs) are located inside an SCC on a negative feedback cycle.

Necessary conditions for an IR variable are, (2) The complementary subsystem to at least one of the acyclic paths from the disturbance variable to the IV should contain a positive cycle (or self-cycle). (3) The complementary subsystem to the positive cycle in one of the complimentary subsystems in condition (2) should not violate conditions 4a, 4b, or 5 discussed below.

Necessary and Sufficient Conditions for a CR variables are, (4) The complementary subsystems to all acyclic paths from the disturbance variable to the CV should each (a) have at least one zero self-cycle (integrator); and (b) not have a cycle containing all of the variables in the subsystem. (5) The complementary subsystems to all nonzero cycles (excluding self-cycles) in each complimentary subsystem in condition 4 should each satisfy conditions 4a, 4b, and 5.

Appendix C

Simulation Results for Robustness Tests

C.1 Simulation Results for Transient Operation

FDI results for the ten faults happening in the transient period are shown in Figures C-1 to C-5.

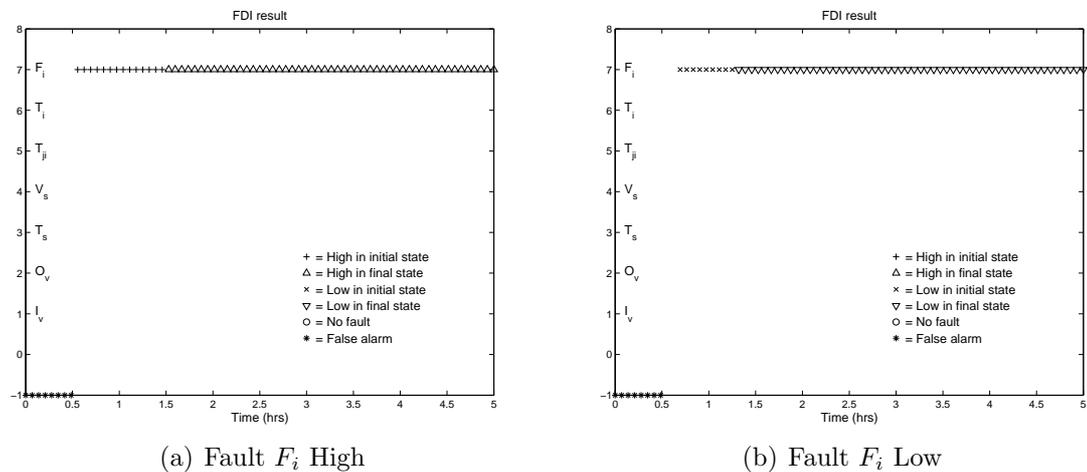
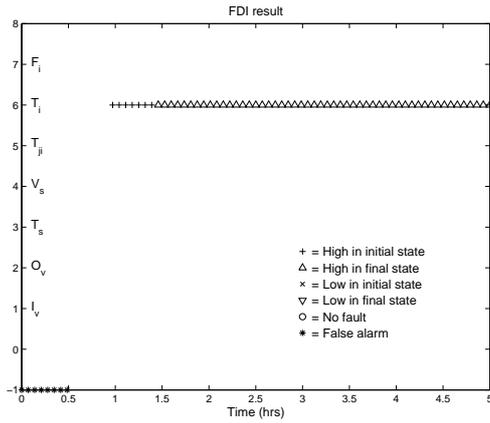
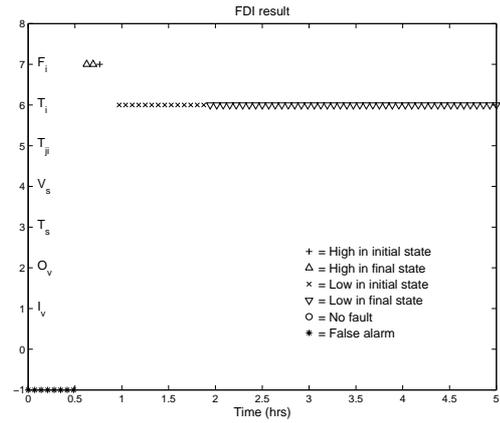


Figure C-1: FDI Results for Fault Happening in a Transient

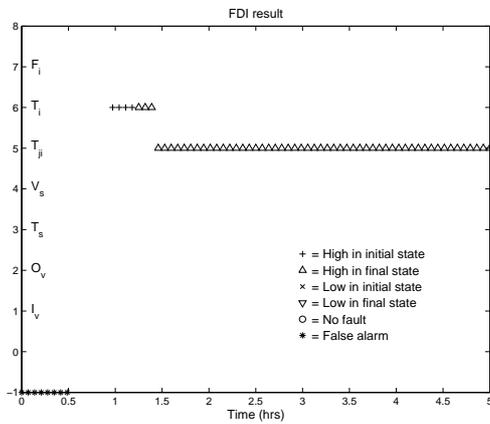


(a) Fault T_i High

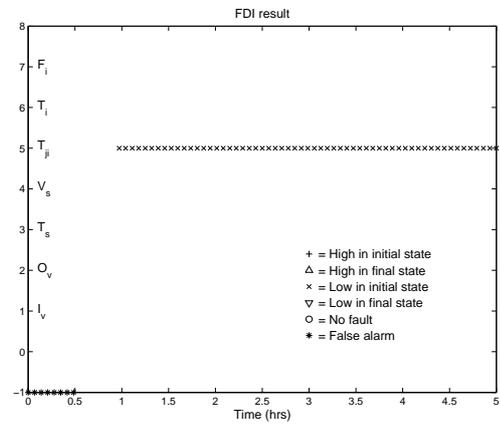


(b) Fault T_i Low

Figure C-2: FDI Results for Fault Happening in a Transient

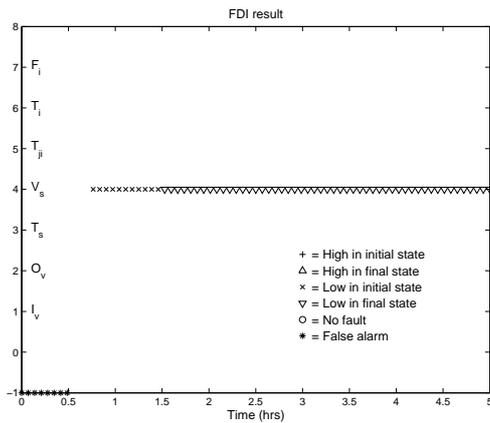


(a) Fault T_{ji} High

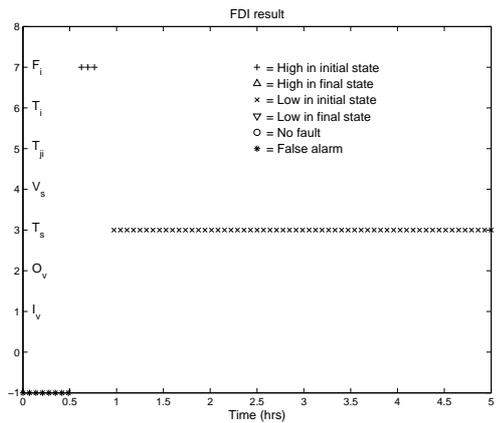


(b) Fault T_{ji} Low

Figure C-3: FDI Results for Fault Happening in a Transient

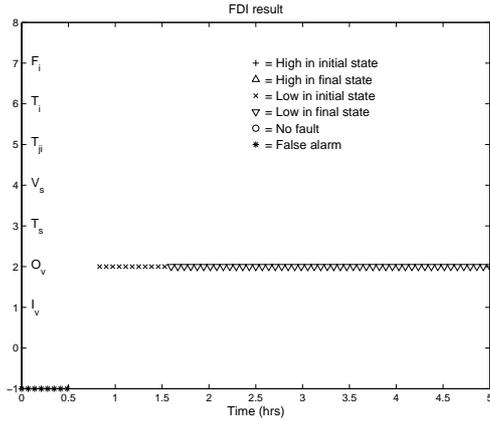


(a) $V_s(-)$ Fault

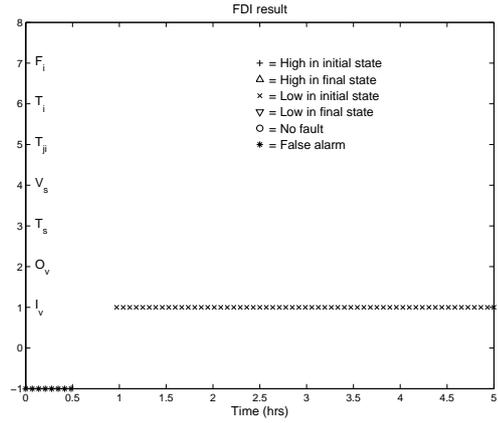


(b) $T_s(-)$ Fault

Figure C-4: FDI Results for Fault Happening in a Transient



(a) $O_v(-)$ Fault

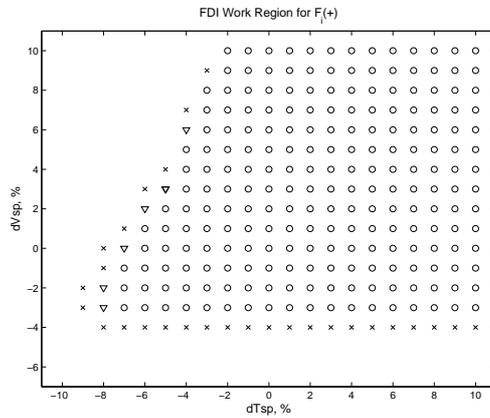


(b) $I_v(-)$ Fault

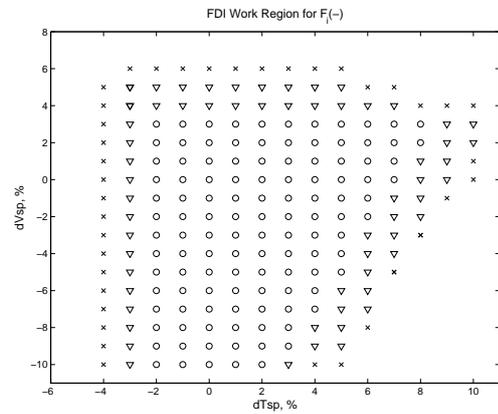
Figure C-5: FDI Results for Fault Happening in a Transient

C.2 Simulation Results for Operation Point Variations

Figures C-6 to C-9 show the other nine faults' workable region. The meanings of the symbols in these figures are: "o", FDI result is correct and fast, less than one hour; "∇", FDI result is correct but late, more than one hour and less than two hours; "x", FDI was not possible, or very late, or only several points (less than 3) are shown.

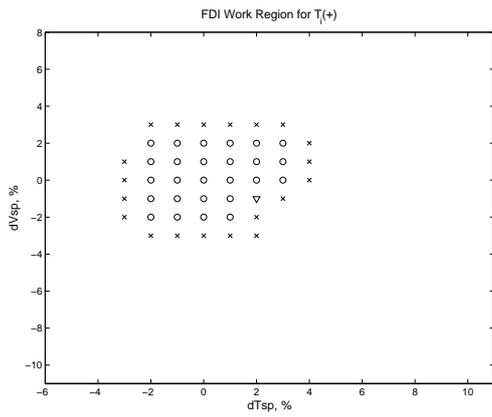


(a) Fault F_i High

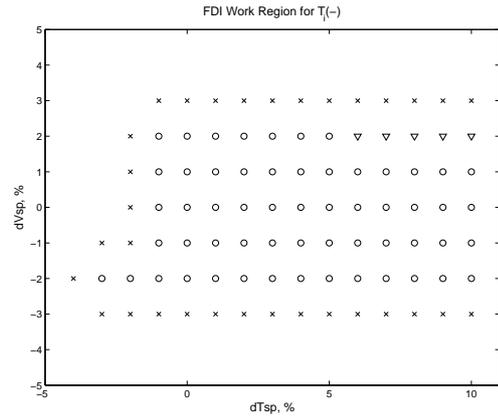


(b) Fault F_i Low

Figure C-6: FDI Work Region

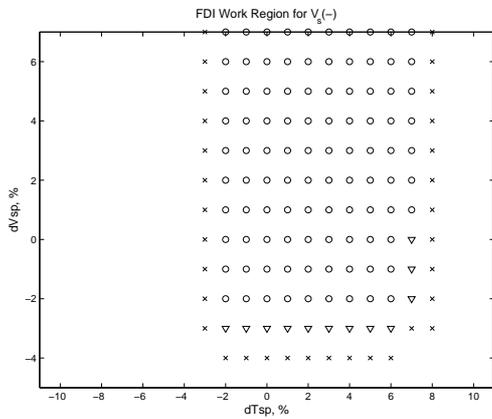


(a) Fault T_i High

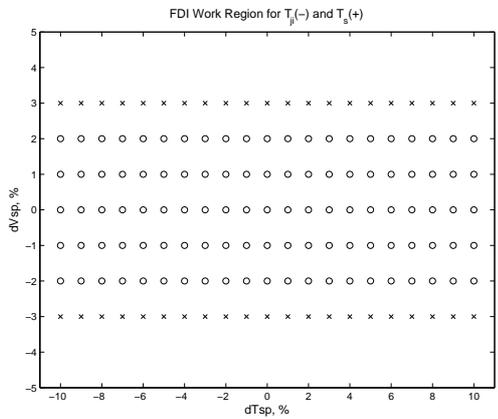


(b) Fault T_i Low

Figure C-7: FDI Work Region

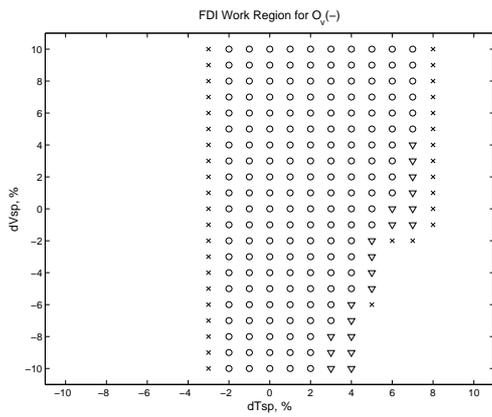


(a) V_s (-) Fault

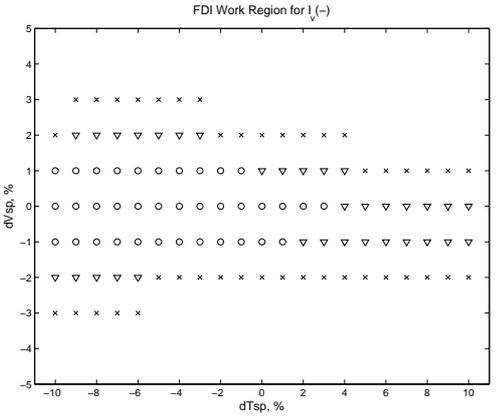


(b) Fault T_s (-) and T_{ji} low

Figure C-8: FDI Work Region



(a) O_v (-) Fault



(b) I_v (-) Fault

Figure C-9: FDI Work Region

C.3 Simulation Result for Operation Fluctuation

For case A, FDI results are given in Figures C-10 to C-14. For case B, FDI results are given in Figures C-15 to C-19. For case C, FDI results are given in Figures C-20 to C-24. For case D, FDI results are given in Figures C-25 to C-29.

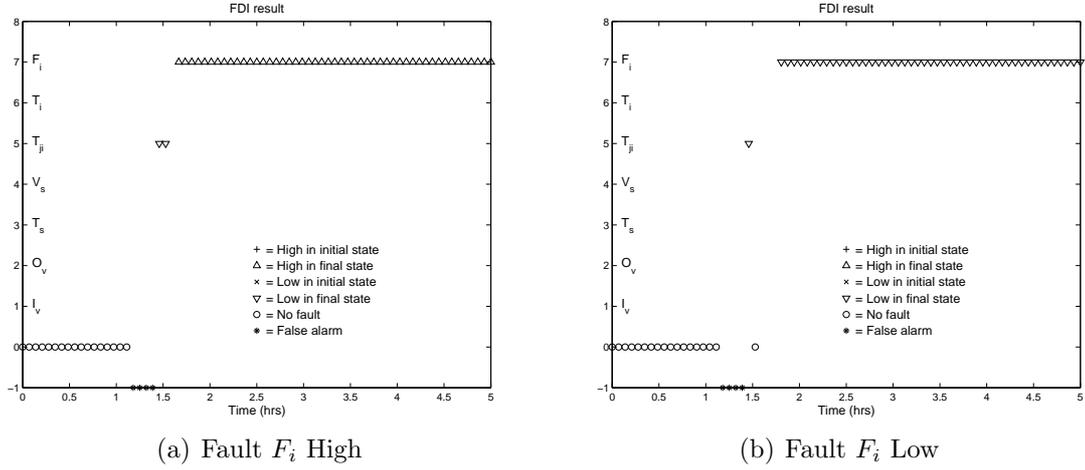


Figure C-10: FDI Results for System with Fluctuation, Case A

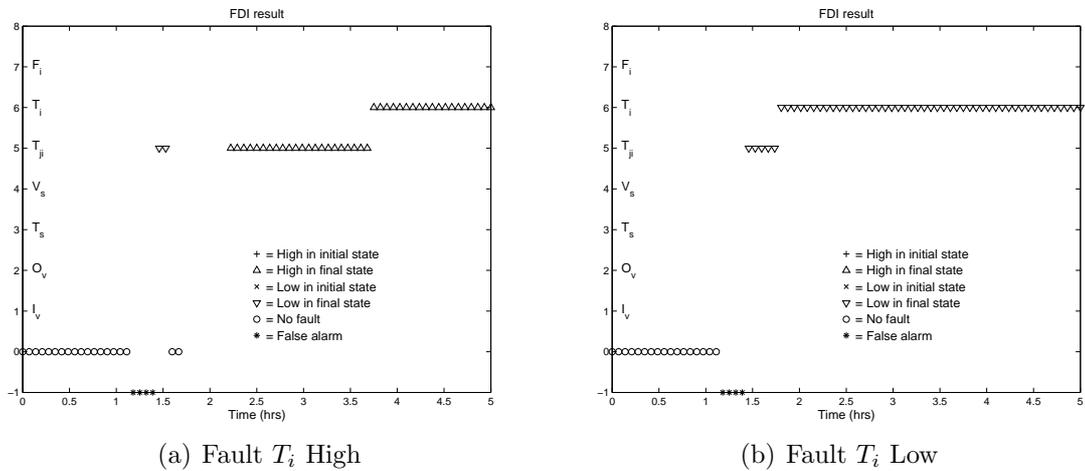
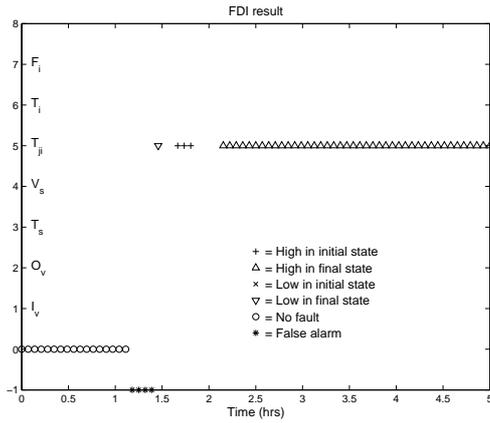
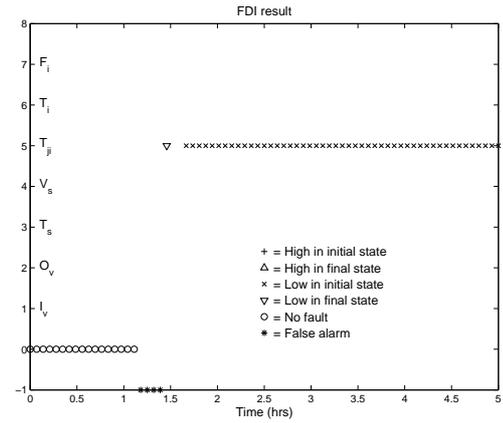


Figure C-11: FDI Results for System with Fluctuation, Case A

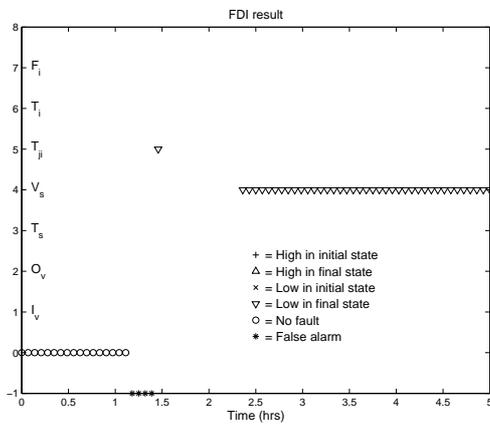


(a) Fault T_{ji} High

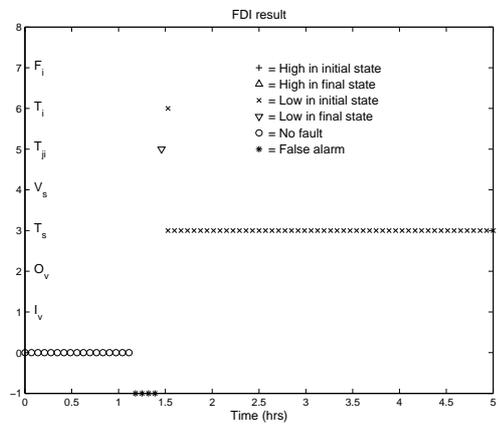


(b) Fault T_{ji} Low

Figure C-12: FDI Results for System with Fluctuation, Case A

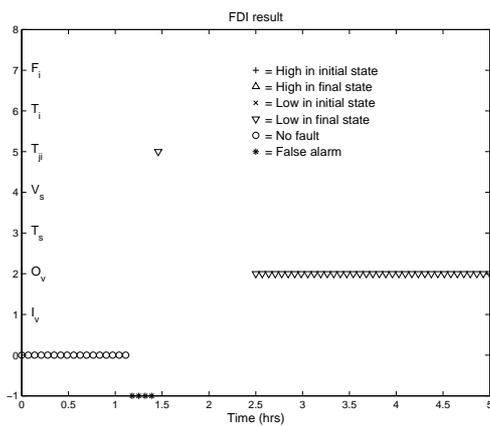


(a) $V_s(-)$ Fault

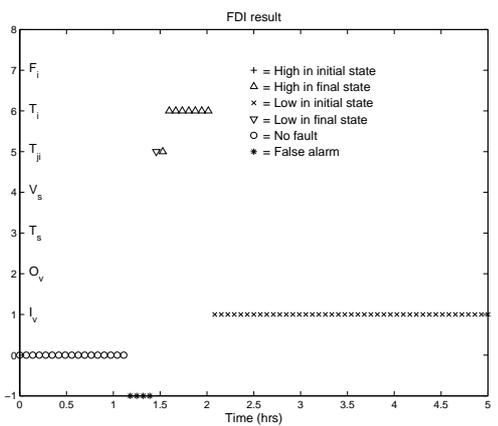


(b) $T_s(-)$ Fault

Figure C-13: FDI Results for System with Fluctuation, Case A

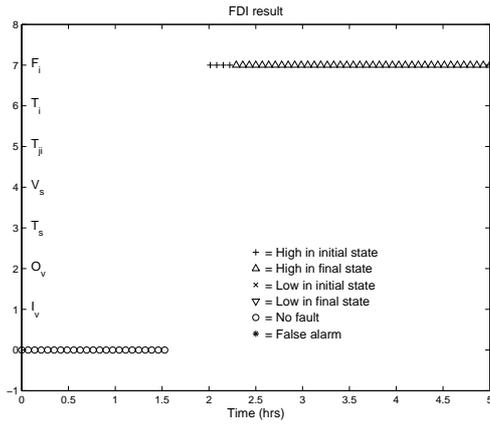


(a) $O_v(-)$ Fault

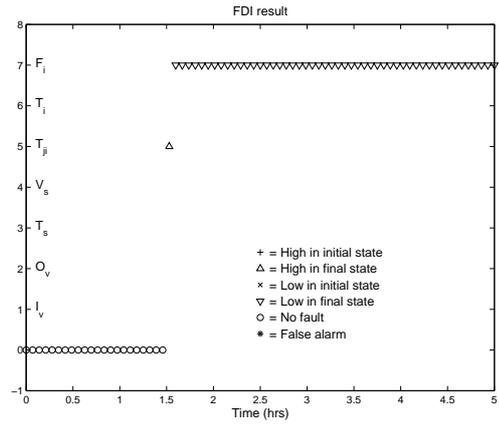


(b) $I_v(-)$ Fault

Figure C-14: FDI Results for System with Fluctuation, Case A

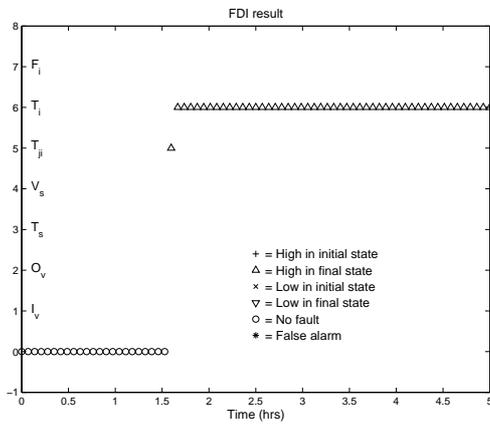


(a) Fault F_i High

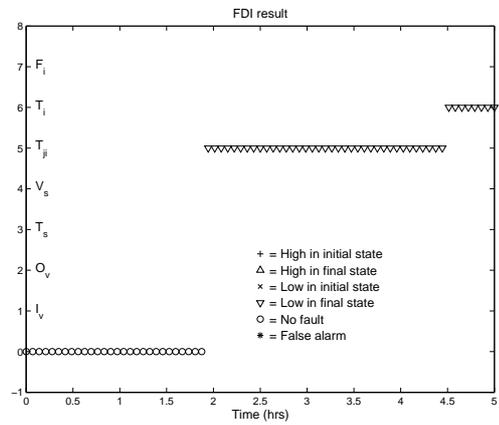


(b) Fault F_i Low

Figure C-15: FDI Results for System with Fluctuation, Case B

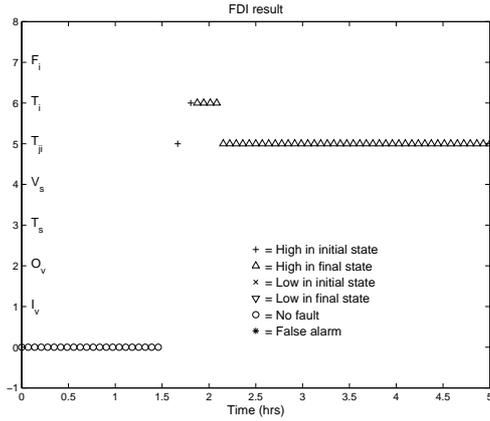


(a) Fault T_i High

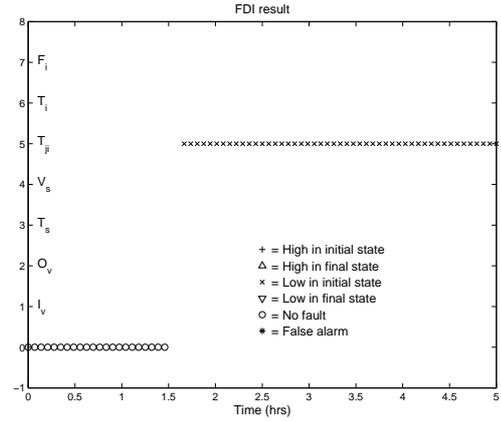


(b) Fault T_i Low

Figure C-16: FDI Results for System with Fluctuation, Case B

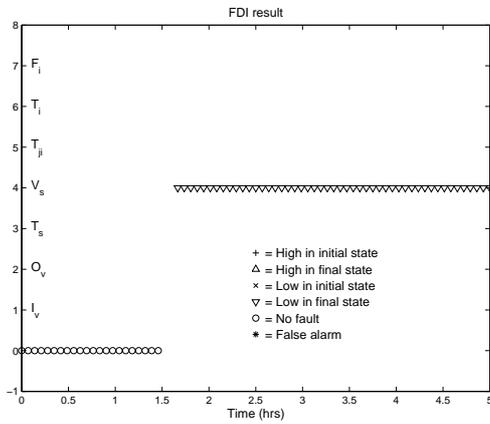


(a) Fault T_{ji} High

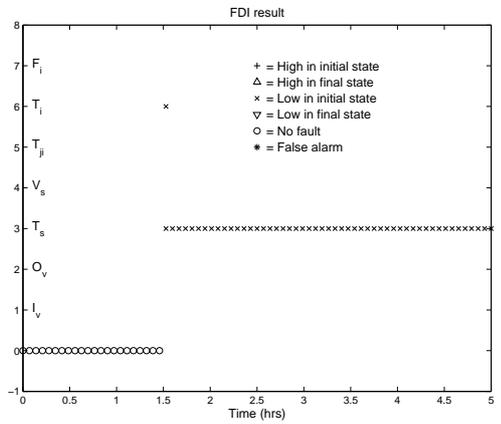


(b) Fault T_{ji} Low

Figure C-17: FDI Results for System with Fluctuation, Case B

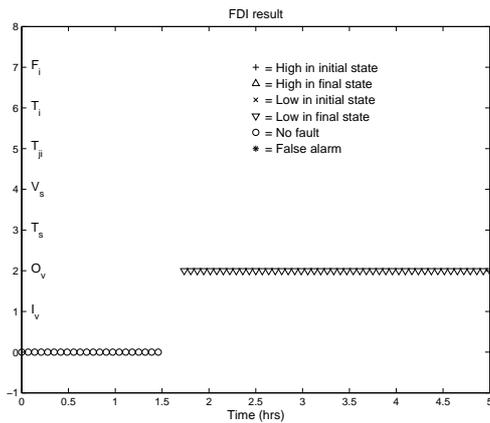


(a) $V_s(-)$ Fault

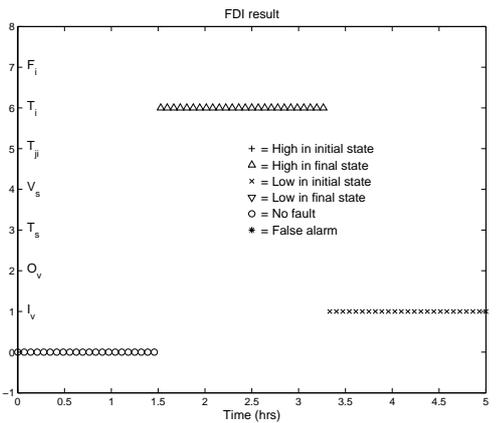


(b) $T_s(-)$ Fault

Figure C-18: FDI Results for System with Fluctuation, Case B

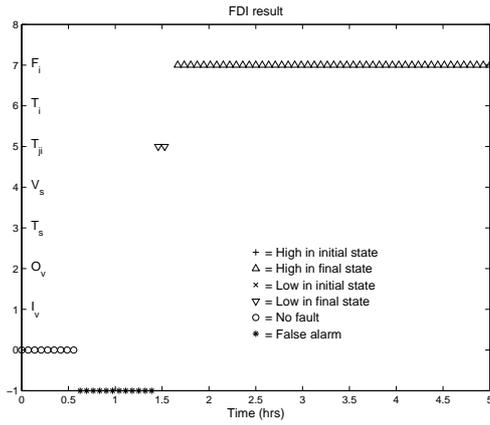


(a) $O_v(-)$ Fault

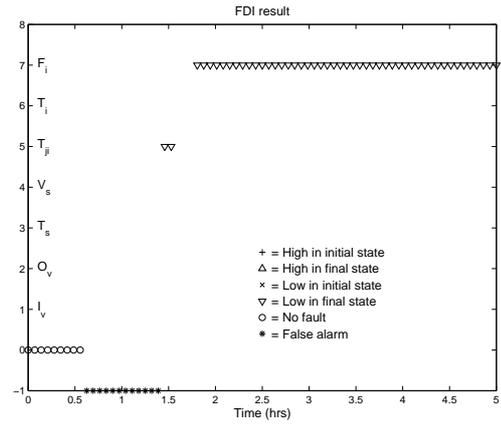


(b) $I_v(-)$ Fault

Figure C-19: FDI Results for System with Fluctuation, Case B

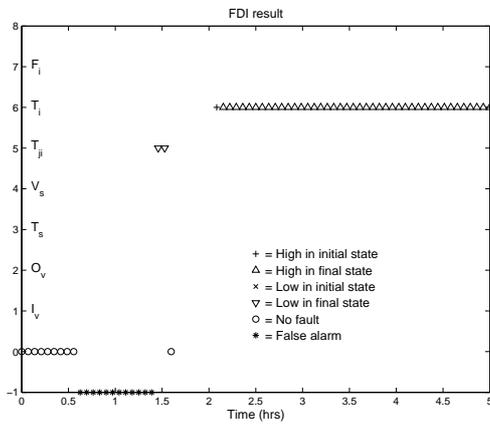


(a) Fault F_i High

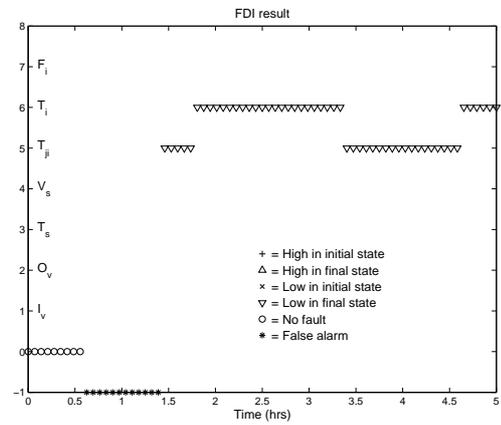


(b) Fault F_i Low

Figure C-20: FDI Results for System with Fluctuation, Case C

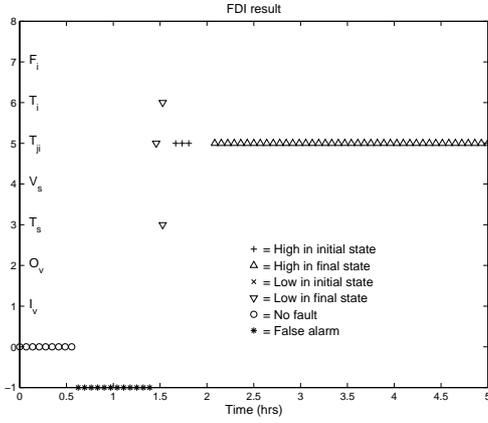


(a) Fault T_i High

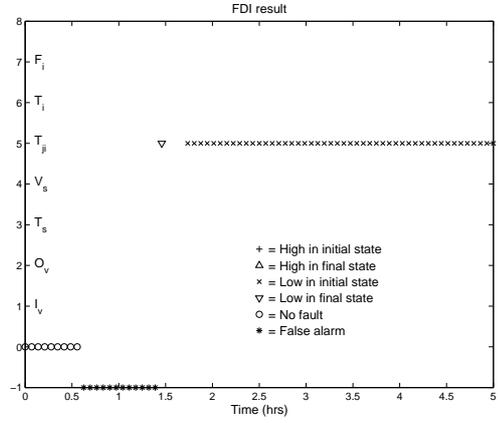


(b) Fault T_i Low

Figure C-21: FDI Results for System with Fluctuation, Case C

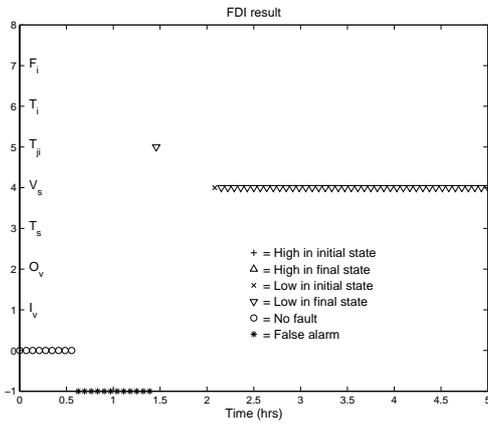


(a) Fault T_{ji} High

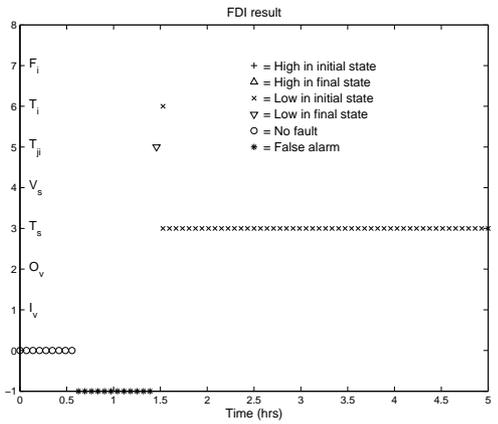


(b) Fault T_{ji} Low

Figure C-22: FDI Results for System with Fluctuation, Case C

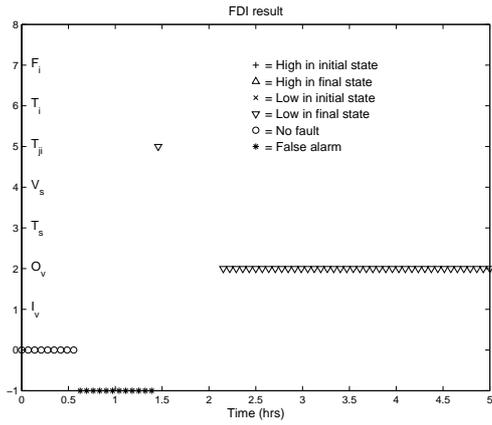


(a) $V_s(-)$ Fault

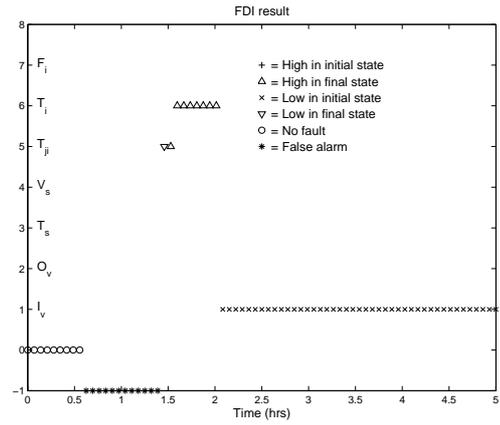


(b) $T_s(-)$ Fault

Figure C-23: FDI Results for System with Fluctuation, Case C

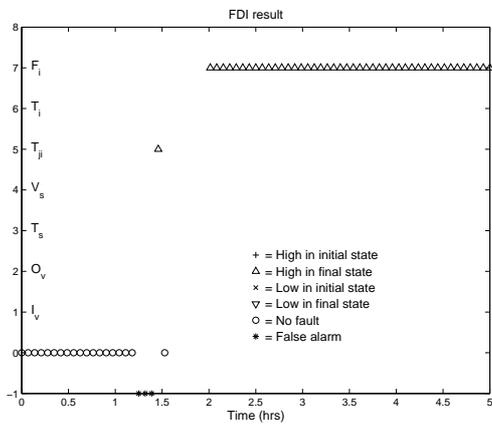


(a) $O_v(-)$ Fault

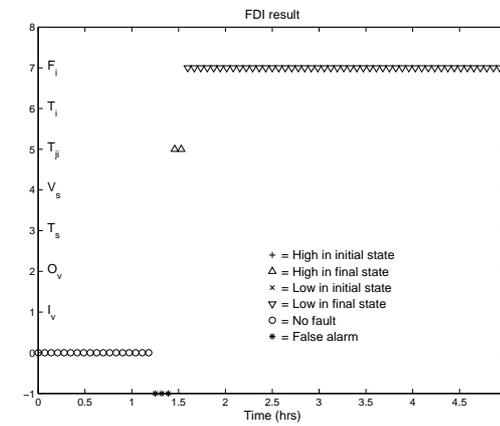


(b) $I_v(-)$ Fault

Figure C-24: FDI Results for System with Fluctuation, Case C

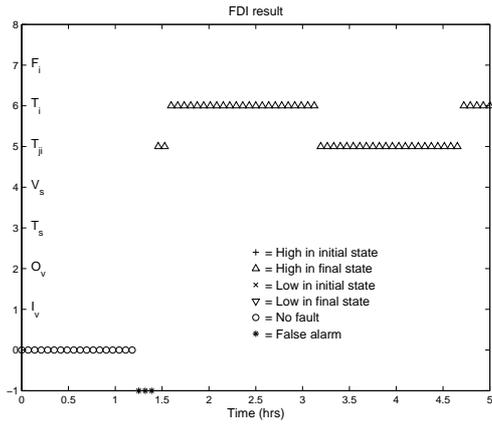


(a) Fault F_i High

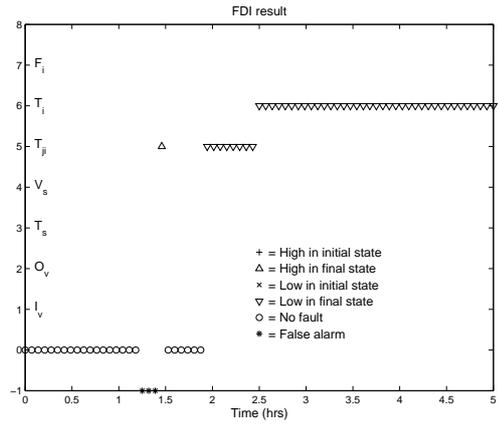


(b) Fault F_i Low

Figure C-25: FDI Results for System with Fluctuation, Case D

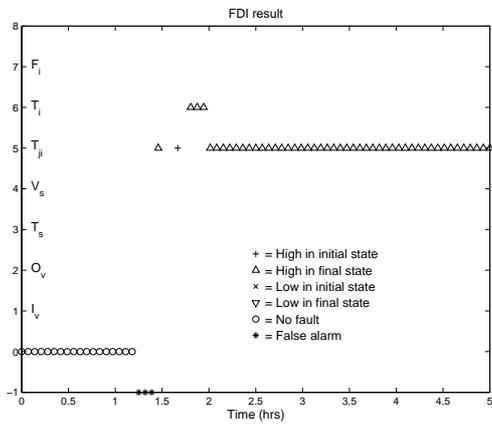


(a) Fault T_i High

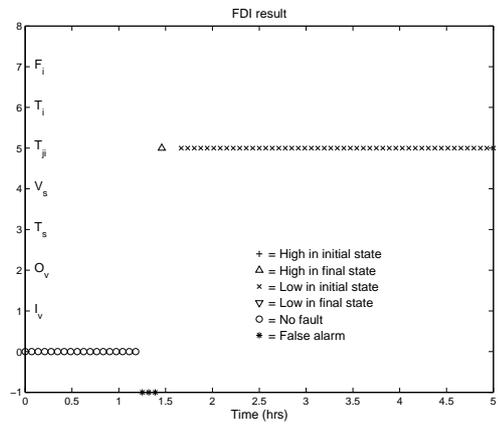


(b) Fault T_i Low

Figure C-26: FDI Results for System with Fluctuation, Case D

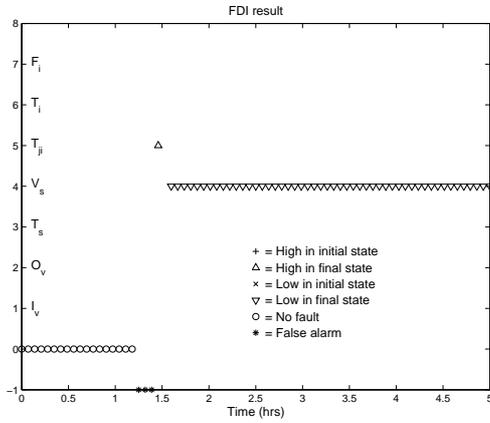


(a) Fault T_{ji} High

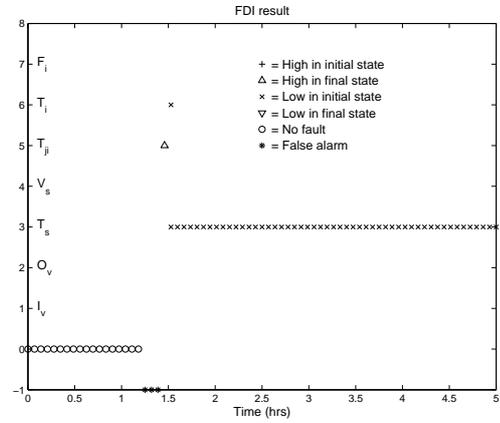


(b) Fault T_{ji} Low

Figure C-27: FDI Results for System with Fluctuation, Case D

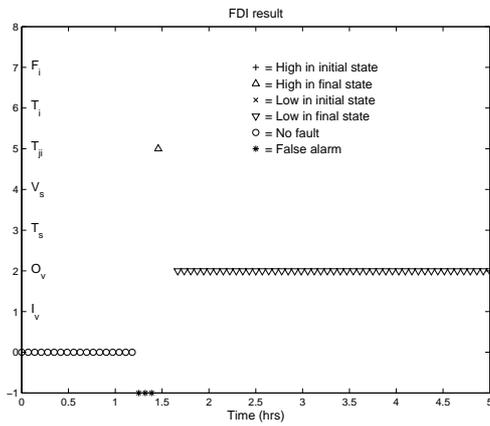


(a) $V_s(-)$ Fault

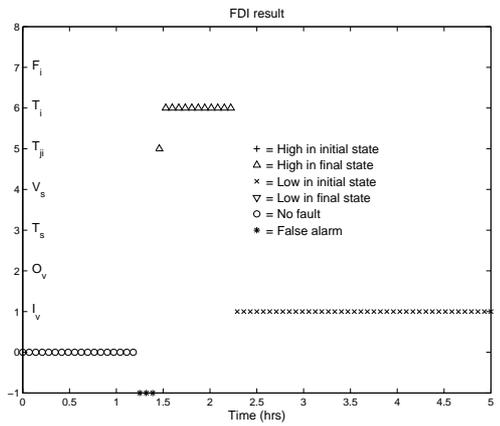


(b) $T_s(-)$ Fault

Figure C-28: FDI Results for System with Fluctuation, Case D



(a) $O_v(-)$ Fault



(b) $I_v(-)$ Fault

Figure C-29: FDI Results for System with Fluctuation, Case D

Appendix D

Matlab Code for Simulation

The FDI algorithm for JCSTR model is listed here. The JCSTR model is used the one supplied by Atalla Sayda [28].

```
% function [P_x] = cstrfdi;

%-----%
%----- Main Script -----%
%-----%

% Step test for Model 1 - CSTR Heater
% Created by John Hedengren
% And modified by Atalla Sayda,
% UNB on May 20, 2004; August 17, 2007

clc; clear all; close all;

global Fault Disturbance t0_fault tf_fault global Vset Tset

% Initial Conditions of States

Fault = 'No Fault'; Disturbance = 'No Disturbance'; Tss
=33.5824+273; Tjss = 104.2784 + 273; Vss = 180; Finss = 0.1; Foutss
= 0.1; Fjinss = 0.15; Tinss = 10 + 273; Tjinss = 120 + 273;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Volume and Temperature Setpoint Modification %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

% To change the setpoints of the volume and temperature,
% Just assign dTset and/or dVset
% to whatever percent change you want to simulate.
% You may use either negative or positive values
% For example, I would assign dTset = + 0.2
% to step up the temperature by 20%.

dTset = 0.0; dVset = 0.0;

% Mixture temperature setpoint (k)
Tset = (1.0 + dTset)*33.5824 + 273;
% Mixture volume setpoint (m^3)
Vset = (1.0 + dVset)*Vss;

% Initial conditions
x0 = [0;Vss;0;Tss;Tjss];

% Simulation time span (sec)
t0 = 0; tf = 5*3600;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Disturbances & Faults      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Introduction of fault scenario into the system      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% In order to apply a fault scenario in the closed loop system,
% Uncomment one fault at a time.
% You may want to modify tfault to change the
%time of fault application into the system.
% tfault should be less or equal to the final time of
%the simulation (tfault <= tf "final time of thesimulation").

% Fault application end time (sec)
t0_fault = 1.5*3600; tf_fault = 5*3600;

% Disturbance = ('Low Mix Inflow');
% Disturbance = ('High Mix Inflow');
% Disturbance = ('Low Inlet Temp');
% Disturbance = ('High Inlet Temp');
% Disturbance = ('Low Heating Fluid Temp');

```

```

% Disturbance = ('High Heating Fluid Temp');
% Fault = ('Faulty Temp Sensor');
Fault = ('Faulty Volume Sensor');
% Fault = ('Faulty Outflow Valve');
% Fault = ('Faulty Heating Fluid Inflow Valve');

% Nonlinear ODE Solution
[t_non,x_non,sys_non] = eufix1y('closnoncstr', t0, tf, x0, 10);
sys_non(1,:) = [Foutss, Fjinss, Finss, Tinss, Tjinss, Vss, Tss,
Tjss];

x_non = x_non(1:25:1801,:); t_non = t_non(:,1:25:1801); sys_non =
sys_non(1:25:1801,:);

% Nonlinear Model Results
Fout_non = sys_non(:,1); Fjin_non = sys_non(:,2); V_non =
sys_non(:,6); T_non = sys_non(:,7) - 273; Tj_non = sys_non(:,8) -
273;

P_non = [V_non,Fout_non,T_non,Tj_non,Fjin_non];

Finsense = sys_non(:,3); Tinsense = sys_non(:,4); Tjinsense =
sys_non(:,5);

% Normal values
Fout_nom = 0.1; Fjin_nom = 0.15; V_nom = 180; T_nom = 33.5824;
Tj_nom = 104.2784;

% % caculating deviation delta = (fault-normal)/(0.002*normal)
V_delt = V_non-V_nom; L_dev = length(V_non); for j=1:L_dev if
abs(V_delt(j)) < 0.001
    V_delt(j) = 0;
end end V_dev = V_delt/(0.002*V_nom); for i=1:L_dev if V_dev(i) >= 1
    V_x(i) = +1;
elseif V_dev(i) <=-1
    V_x(i) = -1;
else
    V_x(i) = 0;
end end Fout_delt = Fout_non-Fout_nom; for j=1:L_dev if
abs(Fout_delt(j)) < 0.001
    Fout_delt(j) = 0;
end end Fout_dev = Fout_delt/(0.002*Fout_nom); for i=1:L_dev if
Fout_dev(i) >= 1

```

```

        Fout_x(i) = +1;
elseif Fout_dev(i) <=-1
        Fout_x(i) = -1;
else
        Fout_x(i) = 0;
end end T_delt = T_non-T_nom; for j=1:L_dev if abs(T_delt(j)) <
0.001
        T_delt(j) = 0;
end end T_dev = T_delt/(0.002*T_nom); for i=1:L_dev if T_dev(i) >= 1
        T_x(i) = +1;
elseif T_dev(i) <=-1
        T_x(i) = -1;
else
        T_x(i) = 0;
end end Tj_delt = Tj_non-Tj_nom; for j=1:L_dev if abs(Tj_delt(j)) <
0.001
        Tj_delt(j) = 0;
end end Tj_dev = Tj_delt/(0.002*Tj_nom); for i=1:L_dev if Tj_dev(i)
>= 1
        Tj_x(i) = +1;
elseif Tj_dev(i) <=-1
        Tj_x(i) = -1;
else
        Tj_x(i) = 0;
end end Fjin_delt = Fjin_non-Fjin_nom; for j=1:L_dev if
abs(Fjin_delt(j)) < 0.001
        Fjin_delt(j) = 0;
end end Fjin_dev = Fjin_delt/(0.002*Fjin_nom); for i=1:L_dev if
Fjin_dev(i) >= 1
        Fjin_x(i) = +1;
elseif Fjin_dev(i) <=-1
        Fjin_x(i) = -1;
else
        Fjin_x(i) = 0;
end end

% fault pattern

P_delt = [V_delt,Fout_delt,T_delt,Tj_delt,Fjin_delt]; P_x =
[V_x;Fout_x;T_x;Tj_x;Fjin_x];

%identify fault

```

```

L = length(t_non);

figure(1) for i= 1:L if P_x(:,i)==[0;0;0;0;0]
    plot(t_non(i)/3600,0,'o'),k=i;
    hold on
end end hold on for i=k:k+1 if P_x(:,i)==[1;1;-1;1;1]
    plot(t_non(i)/3600,7,'+')
elseif P_x(:,i)==[0;1;0;1;1]
    plot(t_non(i)/3600,7,'^')
elseif P_x(:,i)==[1;1;0;1;1]
    plot(t_non(i)/3600,7,'^')
elseif P_x(:,i)==[0;1;-1;1;1]
    plot(t_non(i)/3600,7,'^')
end %% high mix inflow

if P_x(:,i)==[-1;-1;1;-1;-1]
    plot(t_non(i)/3600,7,'x'),
elseif P_x(:,i)==[0;-1;0;-1;-1]
    plot(t_non(i)/3600,7,'v')
elseif P_x(:,i)==[-1;-1;0;-1;-1]
    plot(t_non(i)/3600,7,'v')
elseif P_x(:,i)==[0;-1;1;-1;-1]
    plot(t_non(i)/3600,7,'v')
end %% low mix inflow

if P_x(:,i)==[0;0;+1;-1;-1]
    plot(t_non(i)/3600,6,'+')
elseif P_x(:,i)==[0;0;0;-1;-1]
    plot(t_non(i)/3600,6,'^')
end %% high Inlet Temp

if P_x(:,i)==[0;0;-1;1;1]
    plot(t_non(i)/3600,6,'x')
elseif P_x(:,i)==[0;0;0;1;1]
    plot(t_non(i)/3600,6,'v')
end %% Low Inlet Temp

if P_x(:,i)==[0;0;1;1;-1]
    plot(t_non(i)/3600,5,'+')
elseif P_x(:,i)==[0;0;0;0;-1]
    plot(t_non(i)/3600,5,'^')
end %% High Heating Fluid Temp

```

```

if P_x(:,i)==[0;0;-1;-1;1]
    plot(t_non(i)/3600,5,'x')
elseif P_x(:,i)==[0;0;0;0;1]
    plot(t_non(i)/3600,5,'v')
end
%% Low Heating Fluid Temp

if P_x(:,i)==[-1;-1;1;-1;-1]
    plot(t_non(i)/3600,4,'x')
elseif P_x(:,i)==[-1;-1;0;-1;-1]
    plot(t_non(i)/3600,4,'v')
elseif P_x(:,i)==[0;-1;1;-1;-1]
    plot(t_non(i)/3600,4,'v')
elseif P_x(:,i)==[0;-1;0;-1;-1]
    plot(t_non(i)/3600,4,'v')
% elseif P_x(:,i)==[0;0;0;-1;-1]
%     plot(t_non(i)/3600,4,'v')
end
%% Faulty Volume Sensor

if P_x(:,i)==[0;0;-1;1;1]
    plot(t_non(i)/3600,3,'x')
elseif P_x(:,i)==[0;0;0;1;1]
    plot(t_non(i)/3600,3,'v')
end
%% Faulty Temp Sensor

if P_x(:,i)==[+1;-1;1;-1;-1]
    plot(t_non(i)/3600,2,'x')
elseif P_x(:,i)==[+1;-1;0;-1;-1]
    plot(t_non(i)/3600,2,'v')
end
%% Faulty Outflow Valve

if P_x(:,i)==[0;0;-1;-1;-1]
    plot(t_non(i)/3600,1,'x')
end
%% Faulty Heating Fluid Inflow Valve
hold on end end hold on

for i=k+2:L if P_x(:,i)==[1;1;-1;1;1]
    plot(t_non(i)/3600,7,'+')
elseif P_x(:,i)==[0;1;0;1;1]
    plot(t_non(i)/3600,7,'^')
elseif P_x(:,i)==[1;1;0;1;1]
    plot(t_non(i)/3600,7,'^')
elseif P_x(:,i)==[0;1;-1;1;1]
    plot(t_non(i)/3600,7,'^')

```

```

end                %% high mix inflow

if V_delt(k+1) > V_delt(k+2)
if P_x(:,i)==[-1;-1;1;-1;-1]
    plot(t_non(i)/3600,7,'x'),
elseif P_x(:,i)==[0;-1;0;-1;-1]
    plot(t_non(i)/3600,7,'v')
elseif P_x(:,i)==[-1;-1;0;-1;-1]
    plot(t_non(i)/3600,7,'v')
elseif P_x(:,i)==[0;-1;1;-1;-1]
    plot(t_non(i)/3600,7,'v')
end                %% low mix inflow

end if V_delt(k+1) == V_delt(k+2)
if P_x(:,i)==[0;0;+1;-1;-1]
    plot(t_non(i)/3600,6,'+')
elseif P_x(:,i)==[0;0;0;-1;-1]
    plot(t_non(i)/3600,6,'^')
end                %% high Inlet Temp
end

if T_delt(k+1) > T_delt(k+2)
if P_x(:,i)==[0;0;-1;1;1]
    plot(t_non(i)/3600,6,'x')
elseif P_x(:,i)==[0;0;0;1;1]
    plot(t_non(i)/3600,6,'v')
end                %% Low Inlet Temp
end

if P_x(:,i)==[0;0;1;1;-1]
    plot(t_non(i)/3600,5,'+')
elseif P_x(:,i)==[0;0;0;0;-1]
    plot(t_non(i)/3600,5,'^')
% elseif P_x(:,i)==[0;0;+1;-1;-1]
%     plot(t_non(i)/3600,5,'*')
% elseif P_x(:,i)==[0;0;0;-1;-1]
%     plot(t_non(i)/3600,5,'^')
end                %% High Heating Fluid Temp

if P_x(:,i)==[0;0;-1;-1;1]
    plot(t_non(i)/3600,5,'x')
elseif P_x(:,i)==[0;0;0;0;1]
    plot(t_non(i)/3600,5,'v')

```

```

end                %% Low Heating Fluid Temp

if V_delt(k+1) < V_delt(k+2)
if P_x(:,i)==[-1;-1;1;-1;-1]
    plot(t_non(i)/3600,4,'x')
elseif P_x(:,i)==[-1;-1;0;-1;-1]
    plot(t_non(i)/3600,4,'v')
elseif P_x(:,i)==[0;-1;1;-1;-1]
    plot(t_non(i)/3600,4,'v')
elseif P_x(:,i)==[0;-1;0;-1;-1]
    plot(t_non(i)/3600,4,'v')
% elseif P_x(:,i)==[0;0;0;-1;-1]
%     plot(t_non(i)/3600,4,'v')
end                %% Faulty Volume Sensor
end

if T_delt(k+1) < T_delt(k+2)
if P_x(:,i)==[0;0;-1;1;1]
    plot(t_non(i)/3600,3,'x')
elseif P_x(:,i)==[0;0;0;1;1]
    plot(t_non(i)/3600,3,'v')
end                %% Faulty Temp Sensor
end

if P_x(:,i)==[+1;-1;1;-1;-1]
    plot(t_non(i)/3600,2,'x')
elseif P_x(:,i)==[+1;-1;0;-1;-1]
    plot(t_non(i)/3600,2,'v')
end                %% Faulty Outflow Valve

if P_x(:,i)==[0;0;-1;-1;-1]
    plot(t_non(i)/3600,1,'x')
end                %% Faulty Heating Fluid Inflow Valve
hold on end end hold on,
text(0.1,1,'I_v'),text(0.1,2,'O_v'),text(0.1,3,'T_s'),
text(0.1,4,'V_s'),text(0.1,5,'T_{ji}'),
text(0.1,6,'T_i'),text(0.1,7,'F_i'),

% plot(2.5,2.4,'+'); text(2.6,2.4,'= High in initial state'),
% plot(2.5,2,'^');text(2.6,2,'= High in final state'),
% plot(2.5,1.6,'x'); text(2.6,1.6,'= Low in initial state'),
% plot(2.5,1.2,'v');text(2.6,1.2,'= Low in final state'),
% plot(2.5,0.8,'o');text(2.6,0.8,'= No fault'),

```

```

plot(2.5,7,'+'); text(2.6,7,'= High in initial state'),
plot(2.5,6.6,'^');text(2.6,6.6,'= High in final state'),
plot(2.5,6.2,'x'); text(2.6,6.2,'= Low in initial state'),
plot(2.5,5.8,'v');text(2.6,5.8,'= Low in final state'),
plot(2.5,5.4,'o');text(2.6,5.4,'= No fault'),

axis([0,t_non(L)/3600,-1,8]); xlabel('Time (hrs)'); title('FDI
result','fontsize', 12);

% Plot the results
figure(2);

subplot(3,2,1)
plot(t_non/3600,Vset*ones(size(t_non)),t_non/3600,V_non,'-+'); grid;
title('Measured volume and its setpoint (m3)');

subplot(3,2,2); plot(t_non/3600, Fout_non*1000,'-+'); grid;
title('Mix outflow (i.e., manipulated variable u1) (l/s)');

subplot(3,2,3);
plot(t_non/3600,Tset*ones(size(t_non))-273,t_non/3600,T_non,'-+');
grid; title('Measured mix temperature and its setpoint ( ^{o}C)');

subplot(3,2,4); plot(t_non/3600,Tj_non,'-+'); grid; title('heating
fluid temperature ( ^{o}C)');

subplot(3,2,5); plot(t_non/3600, Fjin_non*1000,'-+'); grid;
title('heating fluid inflow (i.e., manipulated variable u2) (l/s)');

%-----%
%----- CSTR Nonlinear Model -----%
%-----%

function [xdot,sys] = closnoncstr(t,x)

global Fault Disturbance t0_fault tf_fault global Vset Tset

%%%%%%%%%%%%%%
%% Parameters %%
%%%%%%%%%%%%%%

```

```

% Diameter of the reactor (m)
Dr = 5;
% Reactor Height (m)
Hr = 2*Dr;
% Reactor volume (m^3)
Vr = (pi / 2)*(Dr^3);
% Area for heat transfer (m^2)
A = (9/4)*pi*(Dr^2);
% Heat capacity (j/kg.K)
Cp = 4.1868 *1000;
% Mixture inflow 1.5 ft^3/s (m^3/s)
Fin = 0.1;
% Mixture Outflow 1.5 ft^3/s (m^3/s)
Foutss= 0.1;
% Mixture Volume 180 (m^3)
V = x(2);
% Density (kg/m^3)
rho = 997.95;
% Temperature of the mixture feed (K)
Tin = 10 + 273;
% Temperature of the mixture (K)
% Tss = 34.7602 C
T = x(4);
% Heating water inflow 2 ft^3/s (m^3/s)
Fjinss = 0.15;
% Heating water Outflow 2 ft^3/s (m^3/s)
Fjout = 0.15;
% Heating water Volume (m^3)
Vj = 9;
% Temperature of the heating water feed (K)
Tjin = 120 + 273;
% Temperature of the heating water (K)
% Tjss = 103.4932 C
Tj = x(5);
% Heat Transfer coefficient (W/m^2.K)
U = 851.74;
% Temp proportional gain
Kpt = 0.033114;
% Temp integral gain
Kit = 4.5929e-005;
% Volume proportional gain
Kpv = 0.0024;
% Volume integral gain

```

```

Kiv = 1.4621e-006;
% Actuators flags
K1 = 1; K2 = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Disturbances & Faults      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if ((t >= t0_fault) & (t <= tf_fault)),
    switch Fault
        case ('Faulty Temp Sensor'),
            T = 0.8 * x(4) + 0.2 * 273;
            % Temp sensor fault flag
        case ('Faulty Volume Sensor'),
            V = 0.8 * x(2);
            % Volume sensor fault flag
        case ('Faulty Outflow Valve'),
            K1 = 0;
            % Faulty outflow actuator (i.e. stuck valve)
            Foutss = 0.08;
        case ('Faulty Heating Fluid Inflow Valve'),
            K2 = 0;
            % Faulty coolant inflow actuator (i.e. stuck valve)
            Fjinss = 0.12;
    end
    switch Disturbance
        case ('Low Mix Inflow'),
            Fin = 0.1 * 0.8; % Low mix inflow
        case ('High Mix Inflow'),
            Fin = 0.1 * 1.2; % High mix inflow *****
        case ('Low Inlet Temp'),
            Tin = 10 * 0.8 + 273;
            % Low inlet temperature *****
        case ('High Inlet Temp'),
            Tin = 10 * 1.2 + 273; % High inlet temperature
        case ('Low Heating Fluid Temp'),
            Tjin = 120 * 0.8 + 273; % Low coolant temperature
        case ('High Heating Fluid Temp'),
            Tjin = 120 * 1.2 + 273; % High coolant temperature
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%      ODE      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_dot(1,1) = V - Vset; Fout = Foutss + K1*(Kpv*x_dot(1,1) + Kiv *
x(1,1)); if (Fout <= 0.0), Fout = 0.0; end if (Fout >= 0.15), Fout =
0.15; end if (Fin <= 0.0), Fin = 0.0; end if (Fin >= 0.15), Fin =
0.15; end x_dot(2,1) = (Fin - Fout); if ((x(2) <= 0.001) & (x_dot(2,1)
< 0.0)),
    x_dot(2,1) = 0.0;
elseif ((x(2) >= 220.0) & (x_dot(2,1) > 0.0)),
    x_dot(2,1) = 0.0;
end x_dot(3,1) = Tset - T; Fjin = Fjinss + K2*(Kpt*x_dot(3,1) +
Kit*x(3,1)); if (Fjin <= 0.0), Fjin = 0.0; end if (Fjin >= 0.3),
Fjin = 0.3; end A = (pi*(Dr^2)/4) + (4*(x(2))/Dr); x_dot(4,1) =
((Fin*(Tin - x(4))/x(2)) + (U*A*(Tj-x(4))/(x(2)*rho*Cp)); x_dot(5,1)
= (Fjin*(Tjin - Tj)/Vj) - (U*A*(Tj-x(4))/(Vj*rho*Cp)); sys = [Fout;
Fjin; Fin; Tin; Tjin; V; T; Tj];

function [tout, yout, sysout] = eufix1(dyfun, t0, tf, y0, step,
trace)
    %EUFIX1 Solve ordinary state-vector differential
    %equations, low order method.
    %EUFIX1 integrates a set of ODEs ydot = f(y,t) using the most
    %elementary Euler algorithm, without step-size control.
    %
    %CALL:
    %[t, y] = eufix1('dyfun', t0, tf, y0, step, trace)
    %
    %INPUT:
    %dyfun - String containing name
    %of user-supplied problem description.
    %Call: ydot = model(t,y) coded in fname.m => dyfun = 'fname'.
    %t    - Time (scalar).
    %y    - Solution column-vector.
    %ydot - Returned derivative column-vector; ydot = dy/dt.
    %t0   - Initial value of t.
    %tf   - Final value of t.
    %y0   - Initial value column-vector.
    %step - The specified integration step. (Default: step = 1.e-2).
    %trace - If nonzero, each step is printed. (Default: trace = 0).
    %
    %OUTPUT:
    %t    - Returned integration time points (row-vector).

```

```

%y - Returned solution, one column-vector per tout-value.
%
%Display result by: plot(t, y) or plot(t, y(:,2)).

% Initialization
if nargin < 5, step = 1.e-2; end %% default step if not supplied
if nargin < 6, trace = 0; end %% disable trace if not requested
if tf < t0, error('tf < t0!');
return; end %% check for glaring error
t = t0;
h = step;
y = y0(:);
k = 1;
tout(k) = t;
yout(k,:) = y.';
if trace
    clc, t, h, y
end

% The main loop

while (t < tf)
    if t + h > tf, h = tf - t; end
    % Compute the derivative
    [dy,sys] = feval(dyfun, t, y);
    dy = dy(:); sys=sys(:);
    % Update the solution (with no check on error)
    t = t + h;
    y = y + h*dy;
    k = k+1;
    tout(k) = t;
    yout(k,:) = y.';
    sysout(k,:) = sys.';
    if trace
        home, t, h, y, dy
    end
end
if (t < tf) % if true, something bad happened!
    disp('Singularity or modeling error likely.')
    t
end
% ... here is the output (tout in row vector form)
tout = tout(1:k);

```

```

yout = yout(1:k,:);
sysout = sysout(1:k,:);

```

Code for robust test of JCSTR model. Only changes shows here.

```

%%changes made in main fdi code
%%identify fault, changed with original one

L = length(t_non);

%%fault detection
% k=0;
% for i= 1:30
% sum = abs(P_x(1,i))+abs(P_x(2,i))+
%abs(P_x(3,i))+abs(P_x(4,i))+abs(P_x(5,i));
% if sum <2
%     k=k+1;
% end
% end
k=22;

%%changes made in CSTR Nonlinear Model

function [xdot,sys] = closnoncstr1(t,x)

% dTset = 0.02*sin(2*pi*t/(3600*10));
% dVset = 0;

dTset = 0;
dVset = 0.05*sin(2*pi*t/(3600*10));

% Mixture temperature setpoint (k)
Tset = (1.0 + dTset)*33.5824 + 273;
% Mixture volume setpoint (m^3)
Vset = (1.0 + dVset)*180;

```

Vita

Candidate's full name: Liqiang Wang

University attended: Tianjin University, B.Eng, 1997

Publications:

Conference Presentations: