EXPERT SYSTEMS IN ON-LINE PROCESS CONTROL AND FAULT DIAGNOSIS

BY

JIE ZHANG

A THESIS SUBMITTED FOR THE
DEGREE OF
DOCTOR OF PHILOSOPHY

CONTROL ENGINEERING CENTRE
DEPARTMENT OF ELECTRICAL ELECTRONIC
AND INFORMATION ENGINEERING
THE CITY UNIVERSITY
LONDON

JANUARY 1991

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my great gratitude to Professor P. D. Roberts, my supervisor, for his advice, guidance, encouragements, and supports throughout my research work.

I would like equally thank Dr. J. E. Ellis, who is also my supervisor, for his advice and, especially, his assistances in using computational facilities and experimental equipments in the Control Engineering Laboratory.

Finally I would like to thank Mrs. J. Rivellini, secretary of the Control Engineering Centre, and other members of the Centre who assisted me while doing this research.

Declaration

The author grants power of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to him. This permission covers only single copies made for study purpose, subject to normal conditions of acknowledgment.

Publications

The following papers, based on the work described in this thesis, have been published or will be published soon.

An Application of Expert Systems Techniques to the On-line Control and Fault Diagnosis of a Mixing Process, J. Zhang, P. D. Roberts, and J. E. Ellis, *Journal of Intelligent and Robotic Systems,* Vol.1, No.3, 1988, PP209-223.

Fault Diagnosis of a Mixing Process Based on Qualitative Representation of Physical Behaviours, J. Zhang, P. D. Roberts, and J. E. Ellis, *Journal of Intelligent and Robotic Systems,* Vol.3, No.2, 1990, PP103-115.

A Self-learning Fault Diagnosis System, J. Zhang, P. D. Roberts, and J. E. Ellis, to be published in *Transactions of the Institute of Measurement and Control,* 1990.

A Fault Diagnosis System with Self-Reasoning Facilities Based on Qualitative Modelling, J. Zhang, P. D. Roberts, and J. E. Ellis, presented at *IMACS/IFAC International Symposium on Mathematical and Intelligent Models in System Simulation,* Brussels, September 3-7, 1990.

The following papers, based on the work described in this thesis, have been submitted for publication.

On-line Process Fault Diagnosis with Deep Knowledge Based Self-learning of Heuristic Rules, J. Zhang and P. D. Roberts, submitted to *Journal of Intelligent and Robotic Systems,* 1990.

On-line Process Fault Diagnosis Using Neural Network Techniques, J. Zhang and P. D. Roberts, Submitted to *Transactions of the Institute of Measurement and Control,* 1990.

Process Fault Diagnosis with Diagnostic Rules Developed Based on Structural Decomposition, J. Zhang and P. D. Roberts, submitted to *Journal of Process Control,* 1990.

Abstract

In this research expert systems for on-line process control and fault diagnosis have been investigated and the majority of the research is on using expert systems in on-line process fault diagnosis. Several on-line expert systems, including a rule based controller and several fault diagnosis systems, have been developed in this research and are reported in this thesis. The research results obtained demonstrate that rule based controllers can be used in situations where mathematical models for the controlled process cannot be obtained or are very difficult to obtain. The research on on-line fault diagnosis emphasises deep knowledge based approaches. Two avenues in deep knowledge based approaches, namely causal search and qualitative modelling based diagnosis, have been investigated. In the approach of causal search the research results reveal that diagnostic efficiency can be achieved through structural decomposition. A systematic approach for developing diagnostic rules based on structural decomposition is presented in this thesis. Much of the research has been done on qualitative model based fault diagnosis. A qualitative reasoning method which utilizes knowledge on the quantitative relations among variables to reduce ambiguity and can cope with a wider range of situations than Raiman's Order of Magnitude Reasoning is investigated. In the qualitative model based diagnosis the function of the qualitative model is to predict the behaviour of the process under various hypotheses and, therefore, to verify these hypotheses. Further research concerning self-reasoning has been done for the qualitative model based diagnosis approach. Self-reasoning is achieved by backward tracing through the model of the diagnosis system and makes this diagnosis system more intelligent. Self-learning of heuristic rules based on qualitative modelling is investigated and heuristic rules can add efficiency to model based diagnosis. During investigating self-learning of heuristic rules, the good learning property of neural networks is recognised and, neural networks based on-line fault diagnoses are also investigated. The research results reveal that neural networks based diagnosis systems are easy to develop and perform robustly provided that the training data are available.

Chapter 1

Introduction

1.1 Expert systems and their applications in process control

Expert systems have been applied in many areas of process control. More and more applications have been reported recently, which show the great potentials of expert systems in process control. An expert system is a computer program which contains expertise and knowledge about a particular domain and performs some tasks which are usually performed by experts in that domain. With sufficient knowledge, an expert system may even outperform an expert in some situations, since the performance of a human expert is affected by psychological factors, such as boredom, tiredness, and lack of motivation. The expertise of human experts is often accumulated during a long period and, hence, it could be very valuable. Expert system techniques provide a means for exploring and utilizing this valuable knowledge resource.

Expertise in process control engineering includes expertise of process operators related to the operation of a specific process and expertise of control engineers in designing and utilizing different control structures and control algorithms. By making full use of the expertise and knowledge, huge economic profit can result. Good controller performance could lead to good product quality, good supervisory control could reduce energy and raw material consumption, and earlier detection and diagnosis of faults could reduce damage to process equipments and products and reduce the shut down time of the process and, hence, reduce profit losses. The objective of applying expert systems techniques in process control is to make full use of available expertise and knowledge and to achieve economical advantages.

The applications of expert systems in process control can be divided into two categories: off-line applications and on-line applications. Some typical off-line applications include knowledge based control systems design, knowledge based system identification, knowledge based production scheduling, and knowledge based planning. Direct on-line control, on-line fault diagnosis, and supervision are some typical on-line applications. A survey of these applications is provided in the next chapter.

This research aims at investigating the use of expert system techniques in on-line process control and on-line fault detection and diagnosis, and pays more attention to the later. The research on using expert systems in on-line control intends to find and investigate alternative control methods for situations where conventional control methods cannot be efficiently applied instead of replacing the conventional methods in every situation. The research on using expert systems in on-line process fault diagnosis intends to explore more systematic and efficient approaches for building on-line diagnosis systems. Several different expert systems have been developed during this research, and they will be briefly introduced in the following sections.

1.2 A rule based controller

Conventional control algorithms depend on numerical models of the processes to be controlled. However, it may be difficult to obtain the numerical models for some processes. Such processes are usually controlled by human operators, and the operators may have a mental model, in symbolic form, about the process being controlled, and derive control actions from this symbolic model. Expert systems techniques provide a means for dealing with symbolic computation and, therefore, it is possible to develop an expert system which can handle symbolic process models and decide control actions based upon these symbolic models.

The first expert system developed in this research is a rule based controller for a pilot scale mixing process. The rule based controller is developed based on the causal relations inside the process being controlled. These causal relations form a symbolic model of the process and, in some cases, it may be more understandable than any numerical models.

Francis and Leitch (1985a, 1985b) developed an intelligent controller where the system being controlled is similar to the mixing process but is a single-input and single-output (SISO) system. The rule based controller developed here shares some of their ideas, but is an extension to multi-input and multi-output (MIMO) cases. Details of the rule based controller is presented in Chapter 3.

1.3 On-line process fault diagnosis through causal search

The second expert system developed is an on-line fault diagnosis system. The knowledge used for diagnosis is represented by diagnostic rules which are compiled from the knowledge on process unit functions and process system structures. Structural decomposition is used to narrow down diagnosis focus. Based on structural decomposition, fault diagnosis can be performed hierarchically. Through structural decomposition, the process being diagnosed is decomposed into several subsystems and diagnosis is performed by searching for the source subsystem, which is the subsystem where a fault occurs, and locating the fault in the source subsystem. Some researchers (Finch and Kramer 1988, Steels 1989) suggest using functional decomposition to narrow diagnosis search space. In this research, it is demonstrated that structural decomposition can also rapidly focus diagnosis in a small region, and is easier to implement since it corresponds to the plant topology.

A general method is proposed for developing diagnostic rules from the knowledge on process system structures and component functions and this knowledge is represented by several Boolean matrices. Diagnosis systems have been developed for the mixing process and a continuously stirred tank reactor (CSTR) system. Details about this are presented in Chapter 4.

1.4 Fault diagnosis based on qualitative modelling

In the process control domain, process models are sometimes available, and this knowledge can be used in fault diagnosis. For some processes, accurate model parameters may not be available, and in some cases, accurate or direct measurements of some process variables may also be unavailable. The recently developed qualitative modelling techniques (Bobrow 1984) provide a means for solving this issue, since they rely less on accurate measurements and model parameters. Qualitative modelling techniques intend to model process qualitatively, and the qualitative behaviour of a process, such as the directions of deviations of process variables, can be predicted through qualitative simulation.

Based on a qualitative model of a process, fault diagnosis can be done by a procedure of hypothesis formulation and test. When a fault occurs in the process, the actual behaviour of the process will deviate from the predicted one and this can be used to detect the occurrence of a fault. Based on the patterns of violation in the qualitative model and the observed abnormalities, a set of candidate failures (hypotheses) can be formulated. Then these hypotheses are tested on the qualitative model in that the behaviour of the process under these hypotheses is predicted from the qualitative model, and is compared with the actual behaviour. Only the candidate which can explain the observed abnormalities is taken as the diagnosis result. In implementing this diagnosis scheme, it is realised that certain failures, such as sensor failures, should be treated differently from other failures, since the effects of these failures on the process may not be predicted through qualitative simulation. However, these failures can also be diagnosed under the hypothesis-test framework in that they can be discriminated by certain heuristic rules in the test phase. A diagnosis system developed based on this scheme can provide a general framework in that it can be modified for another process by just altering the hypothesis generating scheme, the qualitative models, and some of the specific heuristic rules regarding sensor failures.

A problem associated with qualitative simulation is that ambiguity often occurs due to the lack of quantitative information. In this research, a qualitative reasoning method, which is based on de Kleer and Brown's (1984) confluence based qualitative physics and uses order of magnitude information, is investigated. By taking into account of the available order of magnitude information, ambiguity can be eliminated in some situations. Qualitative modelling based on-line fault diagnosis systems for the mixing process and the CSTR system are developed. The qualitative reasoning method and the two diagnosis systems are described in detail in Chapter 5.

1.5 Qualitative model based diagnosis with self-reasoning facilities

New generation fault diagnosis systems should have the ability to reason their own behaviour and to learn from past experience. With such an objective, some investigations have been performed in building self-reasoning fault diagnosis systems and a self-learning diagnosis system has been developed for the mixing process. It is based on the fault diagnosis system using qualitative simulation described in Chapter 5.

The performance of the diagnosis system described in Chapter 5 is affected by some parameters used in diagnosis. These parameters include the threshold values used in qualitative simulation and parameters used in the diagnosis of sensor failures. Any inappropriate settings of these parameters could result in a wrong diagnosis or miss a fault. Therefore, it is desirable that a fault diagnosis system can reason its own behaviour and find out any inappropriate parameters when it failed to give a desired diagnosis result.

The proposed self-reasoning fault diagnosis system reasons its own behaviour based on its own model and the self-learning is implemented by backward tracing through this model. Once the diagnosis system fails to give a desired result, it will set a hypothesis as its desired output. This output is propagated backwards through the model. Any parameters which are responsible for not giving the desired output are examined. Any inappropriate parameters could be found in such a way. Detailed description about this self-learning diagnosis system is presented in Chapter 6.

1.6 Fault diagnosis using both deep knowledge and heuristic rules with self-learning of heuristic rules

The previously described fault diagnosis systems emphasise the use of deep knowledge. Deep knowledge can provide reliable diagnosis but the diagnosis efficiency may be affected by the deep knowledge based reasoning, since it tries to explore the entire causal path between a fault and the observed abnormalties. Heuristic rules, although they may not give a reliable diagnosis, can usually provide valuable shortcuts in diagnosis since they directly associate symptoms with the corresponding faults. A diagnosis method with the combined use of deep knowledge and heuristic rules is investigated. In this method, heuristic rules are used to propose a hypothesis, while deep knowledge, in the form of qualitative models, is used to confirm this hypothesis. Thus, both efficiency and reliability will be enhanced.

Diagnostic rules may not be perfect, and they may propose wrong hypotheses. Diagnostic rules may also be incomplete and therefore, in some cases, they cannot propose a hypothesis at all. It would be desirable that the diagnosis system can learn heuristic rules itself. Several researchers have been investigating self-learning of heuristic rules. Pazzani (1986, 1987) investigates refining heuristic rules in the situations where existing heuristic rules propose an incorrect hypothesis, and he demonstrated this technique in the diagnosis of the attitude control system of a satellite. Venkatasubramanian and Rich (1989) propose a causality-based failure-driven learning technique which, when the existing heuristic rules propose an incorrect hypothesis, can refine the existing rules and can also learn a new heuristic rule. The two techniques are both failure-driven learning in that learning is initiated when an incorrect hypothesis is proposed. There could exist such situations that no hypothesis can be proposed by the existing heuristic rules. It is desirable that learning could also be initiated in such situations.

A self-learning technique, which takes into account both the situations that an incorrect hypothesis is proposed and that no hypothesis can be proposed, is proposed in this research. It can refine the existing heuristic rules for the first situation, and it can also learn a new heuristic rule. The technique is demonstrated in the fault diagnosis of the mixing process and the fault diagnosis of the CSTR system. Details about this are presented in Chapter 7.

1.7 Process fault diagnosis using neural networks techniques

Several knowledge based diagnosis systems have been briefly introduced so far. These systems provide intelligent assistance to process operators when malfunctions occur in the monitored process. However, the development of these systems may be time consuming and requires certain knowledge and expertise. The self-reasoning and self-learning systems described in Chapter 6 and Chapter 7 perform much better, but they are more complicated. In investigating self-learning of heuristic rules, the good learning property of neural networks is realised. Neural networks have been receiving great attentions recently mainly due to their interesting learning ability and parallel structures. As the final part of this research, the author suggests that neural networks techniques, combined with knowledge based systems, could result in better diagnosis systems, and a technique which uses neural networks for on-line process fault diagnosis is proposed.

A multilayer feed forward neural network is established and is trained from symptom-fault pairs of that process. These training pairs can be obtained from simulation analysis or from past experience on the operation of that process. After training, the network can find out the relations between symptoms and related faults and, can then be used for diagnosis.

The technique is demonstrated by applying it to the fault diagnosis of the mixing process and the fault diagnosis of the CSTR system. It is demonstrated that the neural network based diagnosis systems can diagnose under partial information and partially incorrect information and, furthermore, graceful degradation in performance can be obtained. Details about this are available in Chapter 8.

Chapter 2

A survey of expert system techniques and their application in process control

2.1 Introduction

The applications of expert systems have been dramatically increasing during the last few years. As a mater of fact, one can find huge numbers of reported applications in the periodicals and conference proceedings of many different subjects. An expert system is essentially a computer program which contains expertise and knowledge about a specific domain and performs some tasks which are traditionally carried out by experts in that domain. The early expert systems, such as MYCIN (Harmon and King 1985, Jackson 1986) and DENDRAL (Johnson and Keravnou 1984), contain empirical knowledge of experts in their domains. Many latest expert systems contain knowledge which may not necessarily be experience of some experts, and are also called knowledge based systems. The terms "expert systems" and "knowledge based systems" are used interchangeably in some AI literature (Harmon and King 1985).

Expert systems techniques have been applied in many aspects of process control. These applications can be generally divided into two categories: on-line applications, including direct on-line control, on-line fault diagnosis, on-line supervision; and off-line applications, including control system design, knowledge based system identification, scheduling and planing etc. Expert systems provide a means for utilizing the expertise and knowledge of experienced process operators and control engineers. In on-line process control, expert systems techniques provide an alternative method for controlling some processes where traditional control methods may not be applied efficiently. For example, in a cement production process (King and Karonis 1988), the process is very difficult to model and is subject to large unpredictable disturbances, and conventional control techniques are difficult to apply. Expert systems techniques also provide a means for developing more autonomous systems, integrating control, supervision, and diagnosis.

This chapter is organised as follows: Section 2.2 describes some expert system techniques, a brief description of an expert systems shell: **ExTran**, is given in Section 2.3, some off-line applications of expert systems in process control are briefly introduced in Section 2.4, discussions of expert systems in on-line control and on-line fault diagnosis are presented in Section 2.5 and Section 2.6 respectively. The last section contains a summary of this chapter.

2.2 Expert systems techniques

2.2.1 General structure of an expert system

The architecture of an expert system is shown in Figure 2.1. The knowledge base contains the knowledge about a specific domain, which is supplied by experts or knowledge engineers through the knowledge acquisition subsystem. This knowledge includes general problem solving knowledge as well as specific domain knowledge and is usually in the form of rules and facts. The performance of an expert system is largely determined by the knowledge in its knowledge base, the more knowledge it contains, the more capable it could be. The working memory is used to hold intermediate problem solving results and temporary data about the problem solving state. The inference engine contains the inference strategies and controls that an expert uses when he or she manipulates the facts and rules. The inference engine performs two major tasks. First, it examines existing facts and rules, and adds new facts when possible. Second, it decides the order in which inferences are made. The task of the man-machine interface is to handle all the communications between the user and the expert system. Through this interface and the explanation subsystem, the expert system can explain why and how a particular conclusion is derived.

2.2.2 Knowledge representation

The knowledge in the knowledge base is some facts and rules about a particular domain. Facts describe objects, phenomena, and properties. For example, "Temperature sensor readings change abruptly and randomly" is a fact which describes an observed phenomenon. Human experts can often make decisions with uncertain information. Similarly, facts may also have degrees of uncertainty. For instance, "level in the reactor is high, CF=70%" is an example of inexact facts, where CF is a certainty factor. The relations among facts are described by rules.

There are several different ways to encode the facts and the relationships that constitute knowledge. Some of the commonly used are: semantic networks, object-attribute-value triplets, rules, frames, and logical expressions.

1). Semantic networks. A semantic network is a collection of objects called nodes, which are connected together by arcs or links. Generally, both the links and the nodes are labeled. Figure 2.2 shows a fragment of a knowledge base represented by a semantic network.

Nodes are used to represent objects and descriptors. Objects may be physical objects that can be seen or touched or conceptual entities such as acts, events, or abstract categories. Descriptors provide additional information about objects. Links relate objects and descriptors.

Flexibility is a major advantage of this representational scheme. New nodes and links can be defined as needed. Inheritance is another feature of semantic networks. For example, in Figure 2.2 RLS (Recursive Least Squares method), RPE (Recursive Prediction Error method), and RIV (Recursive Instruments Variable method) will inherit the properties of the node "On-line or recursive methods", namely requiring less memory and suitability for real-time applications.

2). Object-attribute-value triplets. In this scheme, objects may be physical entities or conceptual entities. Attributes are general characteristics or properties associated with objects. The value specifies the specific nature of an attribute in a particular situation. An example representation using O-A-V is shown in Figure 2.3, which states that the level in the reactor is high. In this representation, the object is "reactor", the attribute is "level", and the value is "high". Object-attribute-value triplets are commonly used to represent factual information.

3). Rules. Rules are used to represent causal relations between facts and, are in the following general form

IF Condition(s) **THEN** Conclusion.

Rules and facts can produce new facts. For example, the rule

IF Sensor readings change abruptly and randomly

THEN The sensor failed

and the fact "Temperature sensor 1's readings change abruptly and randomly" will produce the new fact "Temperature sensor 1 failed". A rule is proved (or fired) if its condition part is satisfied. A rule may also have a certainty factor indicating the degree of confidence that the conclusion can be drawn from the given data.

4). Frames. A frame is a description of an object that contains slots for all of the information associated with the object. Slots, like attributes, may store values. Slots may also contain default values, pointers to other frames, sets of rules, or procedures by which values may be obtained. Default values are quite useful when representing knowledge in domains where exceptions are rare. A frame can join together in a single representational strategy two complementary ways to state and store facts: procedural and declarative representations. A declarative representation of a fact is simply an assertion that the fact is true. A procedural representation of a fact is a set of instructions that, when carried out, arrive at a result consistent with the fact. To the degree that facts are independent and changing, then declarative approaches are more understandable or transparent to readers and more easily maintained due to their modularity. Experts and users usually feel more comfortable using a declarative perspective. Procedural representation, on the other hand, is more efficient to use but harder to maintain. The outcome of a procedure is easy to trace, since one can easily examine the flow of instructions. Knowledge engineers are usually more comfortable using a procedural perspective. Frames gain power, generally, and popularity by their ability to integrate both declarative and procedural representations.

5). Logic expressions. Logic provides another way to represent knowledge and the two most common forms of logical notions are propositional logic and predicate calculus. Propositions are statements that are either true or false. Propositions that are linked together with connectives, such as **AND, OR, NOT, IMPLIES,** and **EQUIVALENT,** are called compound statements. Propositional logic is concerned with the truthfulness of compound statements. There are rules for propagating the truthfulness of statements, depending on the connectives. For example, if X is true and Y is false, then the compound statement "X **AND** Y" is false, whereas the compound statement "X **OR** Y" is true.

The elementary unit in predicate logic is an object, and statements about objects are called predicates. For example, "is_high(level_of(tank_1))" is an assertion that says the level of tank 1 is high. Predicates can be linked together by ordinary connectives.

2.2.3 Inference strategies

The inference engine performs two tasks. First, it examines existing facts and rules, and adds new facts when possible. Second, it decides the order in which inferences are made. The most common inference strategy used in knowledge based systems is the application of a logical rule called modus ponens. This rule says that when A is known to be true and if there is a rule "If A then B", it is valid to conclude that B is true. This rule is simple, and hence, reasoning based on it is easily understood. Quite frequently, the information supplied to a knowledge based system is incomplete and uncertain, and some rules may also be uncertain. In such cases, it is required that the knowledge based system should handle uncertain information. This could be done by assigning certainty factors to facts and rules as in MYCIN (Harmon and King 1985, Jackson 1986), and propagating the uncertainty factors during inference. For example, if a conclusion is drawn with a certainty factor X_1, which is in the range $[0, 1]$, and later on the same conclusion is drawn from different facts with a certainty factor X_2, then the certainty for this conclusion is increased to $X_1 + (1 - X_1)X_2$. More advanced technique for handling uncertainty information can be found in (Pearl 1988).

The control portion of the inference engine solves two problems: 1). A knowledge system must have a way to decide where to start; 2). The inference engine must resolve conflicts that occur when alternative lines of reasoning emerge. The control strategies include: backward and forward chaining, depth-first and breadth-first search, monotonic and nonmonotonic reasoning. If the possible outcomes (i.e. the values of the goal attribute) are known, and if they are reasonably small in number, then backward chaining is very efficient. Backward chaining systems are sometimes called goal-directed systems. If the number of possible outcomes is large, a forward chaining strategy would be used. In a forward chaining system, premises of the rules are examined to see whether or not they are true. If they are, then the conclusion are added to the list of facts known to be true and the system examines the rules again. Forward chaining systems are sometimes called data-driven systems. In a depth-first search, the inference engine takes every opportunity to produce a subgoal. A breath-first search sweeps across all premises in a rule before digging for greater detail. Another distinction among inference engines is whether they support monotonic reasoning or nonmonotonic reasoning. In a monotonic reasoning system, all values concluded for an attribute remain true for the duration of the consultation session. Facts that become true remain true, and the amount of true information in the system grows steadily or monotonically. In a nonmonotonic reasoning system, facts that are true may be retracted.

2.3 Some general features of ExTran — an expert systems shell

Since **ExTran** (Razzak, Hassan, and Ahmad 1986) is the expert systems shell that is used in this research, it is briefly introduced in this section.

ExTran, which is short for Expert Translator, is an expert system generator or shell. It is written in Fortran, and therefore, it can be easily linked with external Fortran subroutines. This is suitable for applications, such as expert systems in process control, where some computation is involved. The computation is carried out by external Fortran subroutines. The main characteristics of **ExTran** are listed bellow.

(1). Rule induction.

The knowledge base of expert systems developed by **ExTran** may be expressed as "examples". **ExTran** will then "induce" decision-rules from these examples and use them to build a rule based inquiry system. For example, given the four examples in Table 2.1 about when to use an umbrella, **ExTran** will induce rules shown in Figure 2.4. The "-" in Table 2.1 denotes "don't care", which means that the corresponding attribute is not important. The induced rules can be interpreted as:

```
if weather is wet then
    if in-house is yes then
        don't use umbrella
    else if in-house is no then
        use your umbrella
else if weather is dry then
    don't use umbrella
else if weather is windy
    don't use umbrella
```

Table 2.1: Examples on when to use an umbrella

weather	in-house	decisions
wet	no	use
dry	-	dontuse
windy	-	dontuse
-	yes	dontuse

The system can also accept explicit decision-rules if the knowledge is already available.

(2). Structuring

ExTran provides facilities for flexible decision-rule structuring, which allows hierarchical dependency to be established between decision-rules. Expert systems developed using **ExTran** usually consists of a main problem and several subproblems. Subproblems can be generally divided into class subproblems and attribute subproblems. A subproblem is a class subproblem if it serves as a class value (a conclusion of rules) of the main problem or another subproblem. A subproblem is an attribute subproblem if it serves as an attribute (a condition of rules) of the main problem or another subproblem. In **ExTran** forward and backward chainings are implemented through class and attribute subproblems respectively.

(3). Linkage to external software

An expert system built using **ExTran** may run as a stand-along system or may be a part of a large suite having several external modules. External software may be linked to **ExTran** to capture data, evaluate answers asked by the expert system, execute decisions reached by the system, act as an external utility, etc.

(4). Code generation

ExTran is capable of converting decision-rules into Fortran code and the generated codes are guaranteed to be syntactically correct. This could ease programming effort.

(5). Versatility

Expert systems built by **ExTran** are versatile in the sense that they may be configured to run in various modes. The user can decide on how the questions are to be asked, what text is to be displayed, from where to get the answers, etc.

ExTran is composed of two parts: ACL-Tran, which is short for Analog Concept Learning Translator, and Driver, which is the rule's driver. ACL-Tran is the construction engine of **ExTran**. It enables the developer to define problems, to enter and manage examples, to induce decision rules, to read pre-defined decision rules from files, to test rules against trial data, to convert rules into executable codes, etc. The Driver is a set of object files that should be linked to the developed decision-rules to create the expert system. The procedure for developing expert systems using **ExTran** is illustrated in Figure 2.5, where the dashed lines indicate alternative options.

2.4 Expert systems in process control

Expert systems techniques have been applied in many areas of process control engineering. These applications can be generally divided into two categories: on-line applications and off-line applications. The essential role of expert systems in these applications is an intelligent decision maker, which provides intelligent decisions for encountered situations. Expert systems techniques provide a considerable extension in the applications of computers in process control engineering.

Expert systems in off-line applications generally include: control system design, system identification, production scheduling and planning, training etc. Control system design is a knowledge intensive task and is traditionally carried out by experienced control engineers. The aims of such knowledge based design systems are usually to provide more assistance concerning some tasks which have to be solved by the designer (Lunze 1989), such as planning the design process and execution of a given design plan. The knowledge based system will propose an appropriate design method and sequences of design steps depending upon the properties of the plant and the design specifications. Pang and MacFarlane (1987) describe using expert systems to design multivariable control systems. Rao *et al* (1988) developed an expert system which can determine the optimal control method for a given problem.

With the knowledge about a process, knowledge based identification systems can determine the input signal, system model structure, and the appropriate identification methods. Haest *et al* (1990) developed an expert system, ESPION, which can determine model orders for MISO (multiple input single output) systems. Sanoff and Wellstead (1985) developed an expert identification system which facilitates non-specialists in using adaptive control systems. Betta and Linkens (1990) describe using knowledge based system for dynamic system identification, where the knowledge based system can determine the model structure, choose identification algorithms, and validate the identified model.

Production planning is the process of establishing production rates, work force levels, and on-hand inventories for product families. There exist several optimisation techniques that provide near optimal results for production planning problems. However, these techniques are not widely used by management because: 1), lack of credibility, 2), cost of developing and using models, and 3), excessive data requirements of some models. Most of the planning tasks are performed based on a set of planning rules or guidelines which are formulated from the planner's experience. A knowledge based system can handle such knowledge efficiently and, furthermore, its reasoning procedure is more understandable than other quantitative techniques and, hence, knowledge based systems show their potential in performing production planning tasks. Duchessi and O'Keefe (1990) developed a knowledge based planning system for a company which makes and markets a variety of lawn and gardening products. More such applications can be found in the two special issues of the "Journal of the Operational Research Society" (Doukidis and Paul 1990 a, b).

Expert systems for on-line process control include expert systems for direct on-line control, on-line fault diagnosis and supervision. The aim of this research emphasises on-line applications. Therefore, detailed surveys of expert systems for on-line process control and on-line process fault diagnosis are presented in the next two sections.

2.5 Expert systems for direct on-line control

In such applications, expert systems are used as controllers which derive control actions from measurements, or as parts of controllers which supervise control algorithms. Lunze (1989) refers the former as "heuristic control" and the latter as "expert control". Efstathiou (1989) terms the former as "high **AI**" and the latter as "low **AI**".

2.5.1 Heuristic control

For heuristic control, the knowledge is often represented by rules and therefore, such systems are often called rule based control systems or fuzzy rule based control systems if fuzzy reasoning is adopted. One common feature of such rule based systems is that they do not rely on the numerical models of the processes being controlled. They are used mainly in cases where relatively accurate numerical models cannot be built or are very difficult to build, such as cement kilns (Haspel and Taunton 1986, King and Karonis 1988).

The rules used are in the form

IF Situation **THEN** Control action.

As pointed out by Efstathiou (1986), the inference mechanism used in these rule based control systems is forward chaining since the tasks of these systems are to derive appropriate control actions for different situations and they are data-driven.

One immediate application area of these heuristic control systems is manually controlled processes where the control actions are determined by experienced process operators. In such systems, the experienced operators are replaced by expert systems whose knowledge bases contain the knowledge of experienced process operators. An intelligent controller for the "hot isostotic processing" (HIP) process is described in (Geesey and Blaxton 1988). The HIP process is traditionally manually operated. At the start of a HIP cycle, the operator will set up pressure and temperature schedules designed to produce a final part of some desired density. From on-line measurements, the expert can observe how well the schedule specifications are being met as well as the progress being made in densification of the part inside the chamber and, therefore, he can readjust the temperature and pressure parameters on-line to more accurately control the densification process. The experience of process operators is represented in the knowledge base of an intelligent controller such that the intelligent controller can adjust these parameters automatically or make suggestions on adjustments. The intelligent controller functions as a planning system.

Several intelligent knowledge based controllers for the cement industry have been reported recently (Haspel and Taunton 1986, King and Karonis 1988). The model of a cement kiln is difficult to obtain and, furthermore, the input disturbances are large and unpredictable. Therefore, traditional control methods cannot be applied efficiently. From economic considerations, the process should be operated to maximise production whilst minimising energy consumption. Since an accurate model for a cement kiln can hardly be obtained and the process is subject to a number of significant disturbances, mathematical model based optimal control techniques cannot successfully be applied. However, it is recognised that skilled operators can usually maintain the process in an optimal region. These operators can describe their control actions linguistically as a set of rules. It is demonstrated that by encoding the knowledge of skilled operators as rules and using fuzzy reasoning, the high level supervisory control and optimisation of the kilning stage can be performed automatically by an expert system (Haspel and Taunton 1986, King and Karonis 1988).

Sriada *et al* (1987) describe applications of knowledge based systems in process regulation and servo control. The servo controller implements fast open-loop set-point changes by using two-level bang-bang control. Due to modelling errors and disturbances, the switching parameters cannot be calculated exactly. A knowledge based system is developed to perform simple learning tasks and determine the switching parameters on-line. It is demonstrated that through this simple learning, the knowledge based controller can improve its performance gradually. In the knowledge based regulatory control system, fuzzy heuristic rules are used to deduce control actions. A special group of rules are developed for situations where the process output is near its constraint and, by such means, it is demonstrated that the knowledge based controller makes it possible to operate closer to an output constraint than a conventional PI controller. In many industrial applications, this will achieve economic advantages.

2.5.2 Expert control

The term "expert control" was introduced by Åström (Åström *et al* 1986). The knowledge based element forms a part of the controller, and it determines the appropriate control algorithm for a given situation. The final control action is obtained from the selected control algorithm rather than from the expert system. Expert control involves the construction of a composite control structure for a complex process which includes supervisory functions, adaptive control algorithms, and low level control laws. All of these are managed by an expert system which monitors process parameters and control system performance. In this type of applications, an expert controller might manage the selection and execution of different adaptive control algorithms to maintain the controller parameters at their optimal values for the specific process conditions. In emergency situations, an expert controller may manage the reconfiguration of the controller structure or switch to another more appropriate or robust control algorithm.

One function of the knowledge based elements in these types of applications is to automatically tune a controller (Åström 1989). The tuning knowledge of control engineers is programmed in the knowledge base. Until the present, the most commonly used controller in process control is the PID controller and several researchers have developed different knowledge based systems for tuning PID controller parameters (Lebow and Blankenship 1987, Porter, Jones, and McKeown 1987, McCluskey and Thompson 1987, 1988). In such systems, the characters of the controlled processes are recognised from the transient responses. Controller parameters are determined based on these recognised characters.

To use some of the newly developed control techniques, such as adaptive control, the process operators should have sufficient knowledge related to these techniques as well as experience on using them. However, many process operators may not have the required knowledge and experience, which may account for the reduced popularity of these new techniques. Expert systems can be developed to solve these issues and make these new techniques easier to use. An expert adaptive controller, which can assist process operators in using adaptive controllers, is described by Cooper (1987). The knowledge based component can specify several critical start-up parameters and decide how and when to adjust the forgetting factor, reset the covariance matrix, perturb the process, suspend or restart parameter updating. It can help the control engineer in determining several coefficients in parameter estimation.

Industrial processes are subjected to various operating conditions, including various abnormal conditions. Under different conditions, different controller structures or different control algorithms should be used to achieve the best performance and, furthermore, some abnormal conditions, such as sensor failures, may prohibit certain controller structures. Therefore, it is desirable to have an intelligent controller which can adapt to various operating conditions. An expert adaptive controller for drug delivery systems is presented in (Neat, Kaufman, and Roy 1989), which is developed for the treatment of critically ill patients with cardiac failure in order to reduce the work load of the attending personal. The adaptive control scheme consists of a bank of control algorithms, including a fuzzy controller, a multiple model controller, and a model reference controller, and the co-ordination of these control algorithms and the system stability assessment are orchestrated by a supervisory system. Different controllers are selected for different conditions. An expert multivariable control system for chemical processes is described in (Tzouanas, Luyben, Georgakis, and Ungar 1990a). The expert multivariable controller can select controlled and manipulated variables, determine controller structures, and tune controller parameters for normal operating conditions and various faulty conditions. Applications of this expert multivariable controller to distillation columns are presented in (Tzouanas *et al* 1990b, 1990c).

2.6 Expert systems for on-line process fault diagnosis

One of the first tasks assigned to expert systems in process control is that of process fault detection and diagnosis. This task is a difficult one for process operators, and even well trained operators may have difficulty in diagnosing unanticipated failures, infrequently occurred malfunctions, or incidents where multiple alarms are simultaneously triggered. Therefore, expert diagnosis systems are needed to provide intelligent assistants to process operators. Expert fault diagnosis systems can be divided into shallow knowledge or deep knowledge based approaches according to the nature of the diagnostic knowledge employed.

2.6.1 Shallow knowledge based diagnosis

Shallow knowledge based diagnosis systems capture the relations between observed abnormalities and the associated malfunction. The knowledge used is the empirical associations between symptoms of a fault and the fault itself, and is acquired from process operators. The knowledge is represented by rules and, quite often, uncertain reasoning is used since the knowledge is frequently uncertain. These diagnosis systems are similar to MYCIN (Harmon and King 1985, Jackson 1986), which is a typical shallow knowledge based expert medical diagnosis system capable of handling uncertain information.

A key task associated with the shallow knowledge based diagnosis systems is knowledge acquisition. Expertise covering a wide range of malfunctions must be encoded into the expert system. The knowledge requirements are unstructured and may be broad in scope. The task of knowledge acquisition is very time consuming since the process operators may know little about knowledge engineering and, therefore, the interchange of information between a knowledge engineer and a process operator may not be carried out efficiently. This issue is often referred to as the "knowledge engineering bottle neck" (Moor and Kramer 1986, Price and Lee 1988). The knowledge base is highly specific to the particular plant and there is no guarantee that it is complete. In an industrial process, many faults needing to be diagnosed may never have been experienced and, for new or recently developed plants, there may be little applicable experimential knowledge.

Due to these drawbacks, the shallow knowledge based diagnosis is often applied to a process where model based reasoning cannot be applied, or applied to small scale processes where the knowledge required for diagnosis is limited. Shallow knowledge in the form of heuristic rules can usually provide valuable short cuts in diagnosis since the rules associate symptoms directly with the corresponding malfunctions. Therefore, shallow knowledge is often combined with, and supplements, deep knowledge based diagnosis schemes. A diagnosis scheme which integrates deep knowledge and shallow knowledge is described in (Venkatasubramanian and Rich 1988). Lapointe *et al* (1989) developed an expert diagnosis system for a waste water treatment process — BIOXPERT, where shallow knowledge is used to diagnose the frequently occurred faults.

2.6.2 Deep knowledge based diagnosis

The so called deep knowledge includes models of the process being diagnosed and faulty models of different process units. The model of a process can be in various forms. It can be in the form of a set of numerical equations, or a qualitative model, or even in the form of rules compiled from a model. As suggested in (Scarl, Jamieson, and Delaune 1987), diagnosis systems based on any type of models, regardless of the depth of the models, can be called deep knowledge based systems. Based on the deep knowledge about a process, diagnosis can be performed more reliably.

There are several different approaches in deep knowledge based diagnosis. Some of the commonly used are causal search, diagnosis based on numerical model equations, and diagnosis based on qualitative modelling.

(1) Causal search. The diagnosis system attempts to trace the observed abnormalities to their origin. The knowledge used is the descriptions of unit functions and system structures information which includes the connectivity of different units. From this knowledge, causal paths between a fault and observed abnormalities can be established. Fault diagnosis of electronic and digital circuits typically employs this method (Davis 1983, 1984).

An efficient technique for representing causality relations among process variables is the Signed Directed Graph (SDG) (Iri *et al* 1979). The SDG is used to represent pathways of causality in the fault-free process. The nodes of the SDG correspond to state variables, alarm conditions, or failure origins, and the edges represent the causal influences between the nodes. The directions of the deviations of the nodes are represented by the signs on the branches, + or −, indicating that the cause and effect variables tend to change in the same or opposite directions respectively. The earlier diagnosis systems based on SDG do not use expert system techniques (Iri *et al* 1979, Shiozaki *et al* 1985). Recently, several researchers have attempted to formulate diagnostic rules from the SDG representation of processes. Kramer and Palowitch (1987) demonstrate that diagnostic rules can be derived from the SDG representation and that fault diagnosis based on these rules is more efficient.

Based on the knowledge of system structure and component functions, fault diagnosis can be performed hierachically. The process being diagnosed can be decomposed, either functionally or structurally, into several subsystems and, therefore, diagnosis can be rapidly focused into a small region (Finch and Kramer 1988, Shum *et al* 1988, Steels 1989).

(2) Diagnosis using numerical model equations. In the process control domain, a model of the process and various constraints derived from mass and energy balance in the form of numerical equations are usually available. These equations, also called governing equations (Kramer 1987), provide important information about the process, and can be used in diagnosis. Due to measurement noise, unmeasured disturbances, and inaccuracies in certain parameters of these equations, there exist equation residuals. During normal operation, the equation residuals should all be within their tolerances. Once a fault occurs in the process, some equation residuals will deviate from their tolerances. By analysing these residuals, a fault may be diagnosed. Several diagnosis methods based on numerical equations are reported recently (Kramer 1987, Lutcha and Zejda 1990, Petti, Klein, and Dhurjati 1990).

The governing equations based diagnosis are briefly summarised here. Let C_i^+, C_i^-, and C_i^0 be the conditions for positive and negative constraint violations and constraint satisfaction of the *ith* constraint (governing equation) respectively. Let F be the set of all possible faults with members f. The set of faults that are sufficient to cause violation of the *ith* constraint are defined as follow:

$$H_i^+ = \{\forall f, f \rightarrow C_i^+\}$$

$$H_i^- = \{\forall f, f \rightarrow C_i^-\}$$

Let the condition of the plant be C^*, where $C_i^* = C_i^+, C_i^-$, or C_i^0 depending on whether the *ith* constraint is violated positive, negative, or satisfied. Let H_i^* be the fault set activated by the condition of the *ith* constraint, then

$$C_i^* = C_i^+ \rightarrow H_i^* = H_i^+$$

$$C_i^* = C_i^0 \rightarrow H_i^* = -(H_i^+ \cup H_i^-)$$

$$C_i^* = C_i^- \rightarrow H_i^* = H_i^-$$

For the case of a single fault, viable single fault hypotheses are those that account for all violated constraints. Therefore, the set of single-fault hypotheses are

$$S = (H_1^* \cap H_2^* \cap \cdots \cap H_{NC}^*)$$

where NC is the number of constraints. Formulae for resolution of multiple faults are given in (Kramer and Palowitch 1985).

Based on numerical equations, non-Boolean reasoning can be applied and, therefore, graceful degradations in performance can be obtained (Kramer 1987). It is demonstrated that through non-Boolean reasoning, the diagnosis system will not be sensitive to measurement noise.

(3) Diagnosis based on qualitative modelling. The above described numerical equation based diagnosis method may not be suitable for a process where accurate measurements or direct measurements of some process variables are not available, or some model parameters are not known accurately. For such situations, qualitative modelling techniques (Bobrow 1984) can be used in diagnosis. The qualitative model of a process is often obtained from its quantitative model and, therefore, it can correctly describe the process. Through qualitative simulation, the deviations of certain process variables can be obtained.

Qualitative simulation based diagnosis is usually performed through the hypothesis-test strategy (Moor and Kramer 1986). Because qualitative simulation can predict the deviations of certain process variables under normal operating conditions as well as various faulty conditions, diagnosis can be done by first formulating a set of hypotheses, and then testing these hypotheses using the qualitative model; the hypothesis which can explain the observed abnormalities is the diagnosis result.

Several researchers have been investigating using qualitative modelling in process fault diagnosis. Qualitative modelling of chemical processes is investigated by Oyeleye and Kramer (1988) and Waters and Ponton (1989). Herbert and Williams (1986, 1987) investigated using qualitative modelling in the diagnosis in power plant. The author has performed research in qualitative simulation incorporating order of magnitude information, and using qualitative simulation in on-line process fault diagnosis. These will be described in detail in Chapters 5, 6, and 7.

A problem associated with qualitative modelling is that ambiguity often occurs due to the lack of quantitative information. Ambiguity prevents further discrimination of a set of plausible hypotheses, which could be discriminated with a detailed quantitative model. Several approaches have been investigated to reduce ambiguity. Raiman (1986) investigates using order of magnitude information among variables to reduce ambiguity. Oyeleye and Kramer (1988) show that additional qualitative constraints could be derived from redundant numerical equations and ambiguity could be reduced.

2.7 Conclusions

In this chapter expert systems and their applications in process control are briefly introduced. The basic structure of an expert system, various knowledge representation schemes and inference strategies are presented. Some general features of **ExTran**, an expert systems shell used in this research, is also briefly described in this chapter. A review of applications of expert systems in process control, especially in on-line process control and fault diagnosis, is provided.

Expert systems for on-line process control can be generally divided into "heuristic control" and "expert control" according to the roles of expert systems. Heuristic control can be used to automate some manually controlled processes which are difficult to be controlled by conventional methods. Expert control is generally used to provide some supervisory functions for conventional control algorithms, such as controller parameter tuning, determining controller structure, and to assist process operators in using advanced control techniques, such as adaptive control. Expert systems for on-line process fault diagnosis can be generally divided into a shallow knowledge based approach and a deep knowledge based approach according to the knowledge used. The shallow knowledge based approach is generally used in small scale processes or in some processes where a deep knowledge based approach cannot be applied. Shallow knowledge is often used to supplement deep knowledge to improve diagnostic efficiency. Deep knowledge based approaches can usually provide reliable diagnosis for a wide range of faults.

The discussion in this chapter provides an environment encompassing the research of this thesis. The research on on-line process fault diagnosis provided in this thesis aims to develop, investigate and explore more systematic, more efficient, and more reliable fault diagnosis methods.

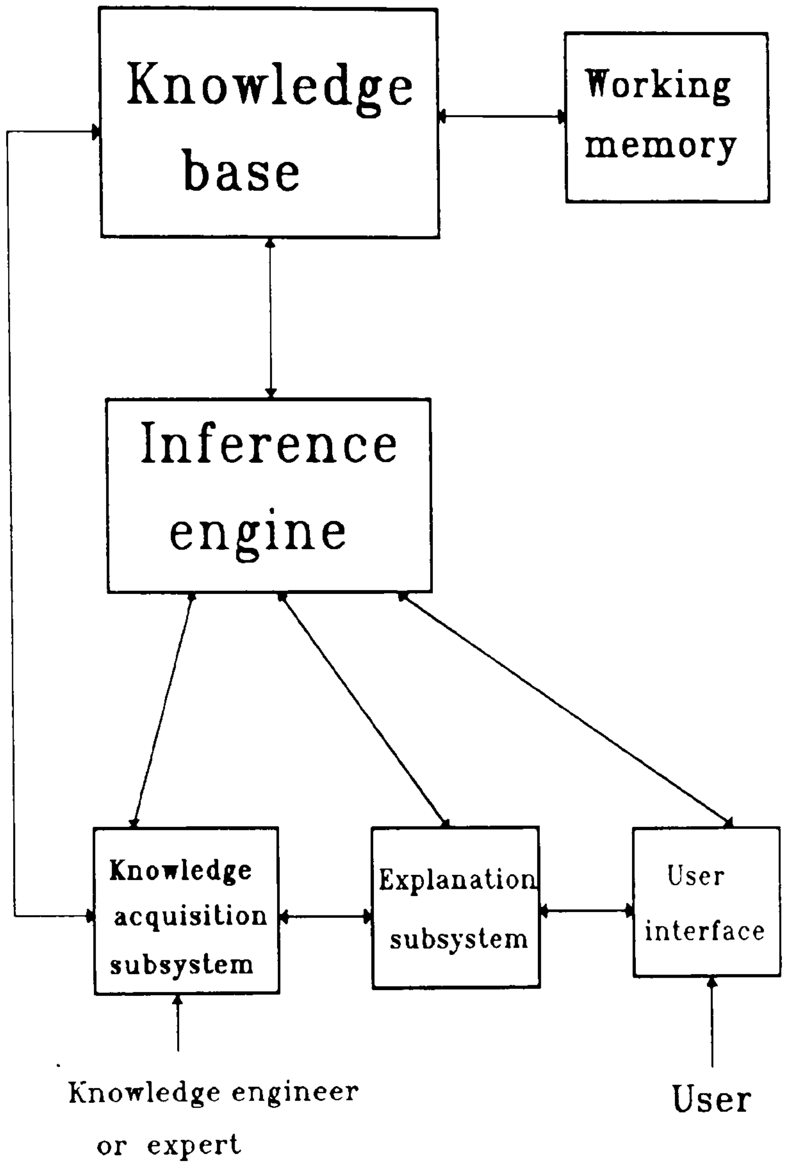


Fig 2.1 The architecture of an expert system

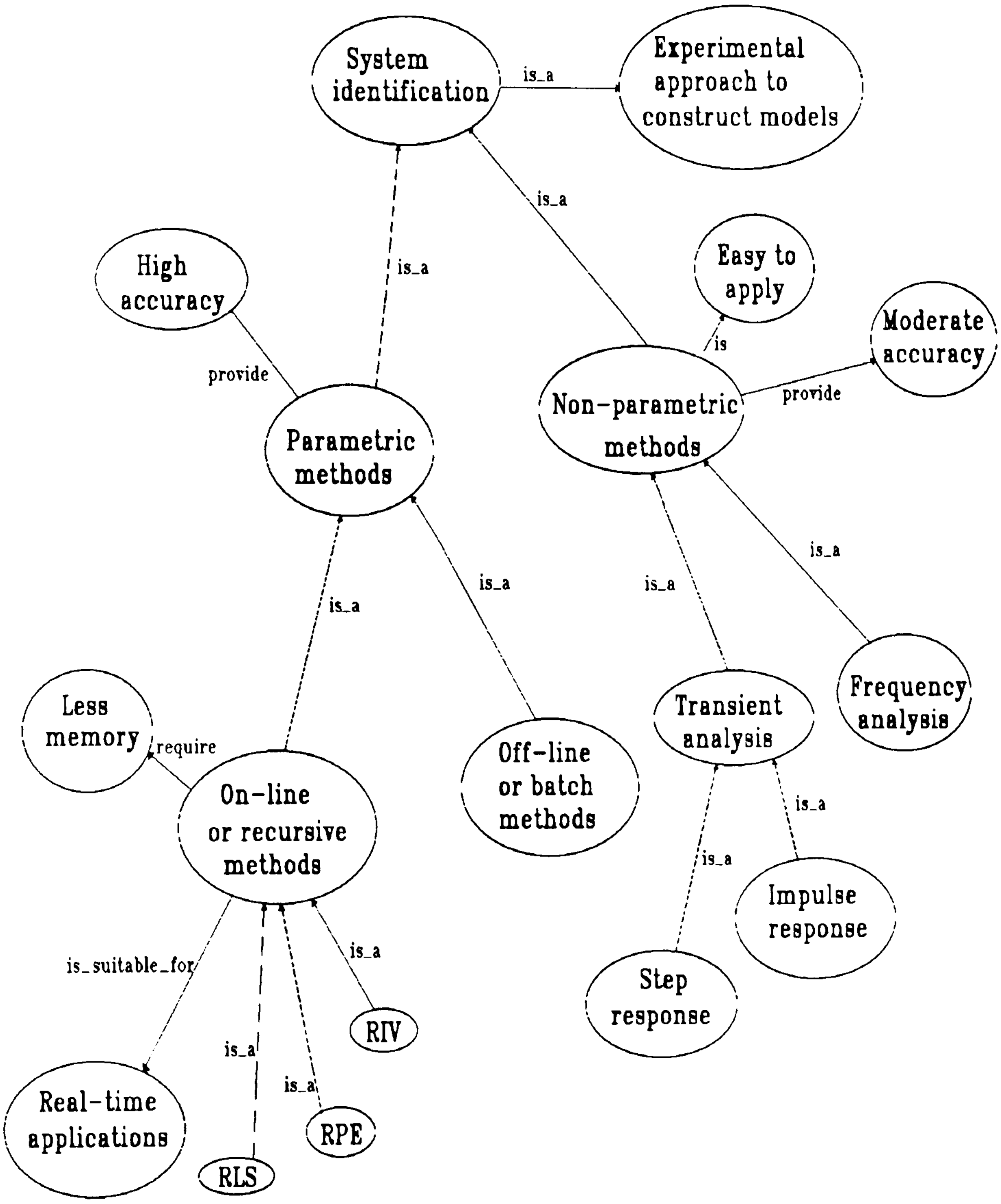


Fig.2.2 Fragment of a semantic network

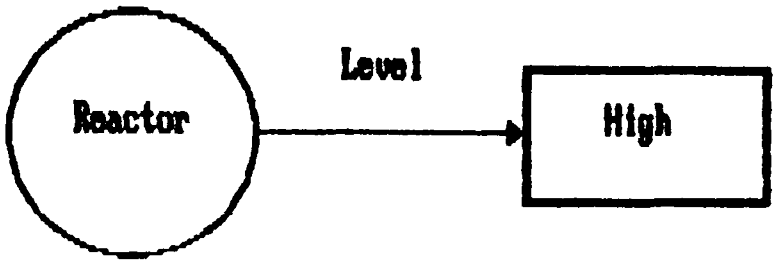


Fig. 2.3 An object-attribute-value triplet

```
[ weather] :
     wet : [ in_house] :
               yes : dontuse
               no : use
     dry : dontuse
   windy : dontuse
```

Figure 2.4 ExTran induced rules

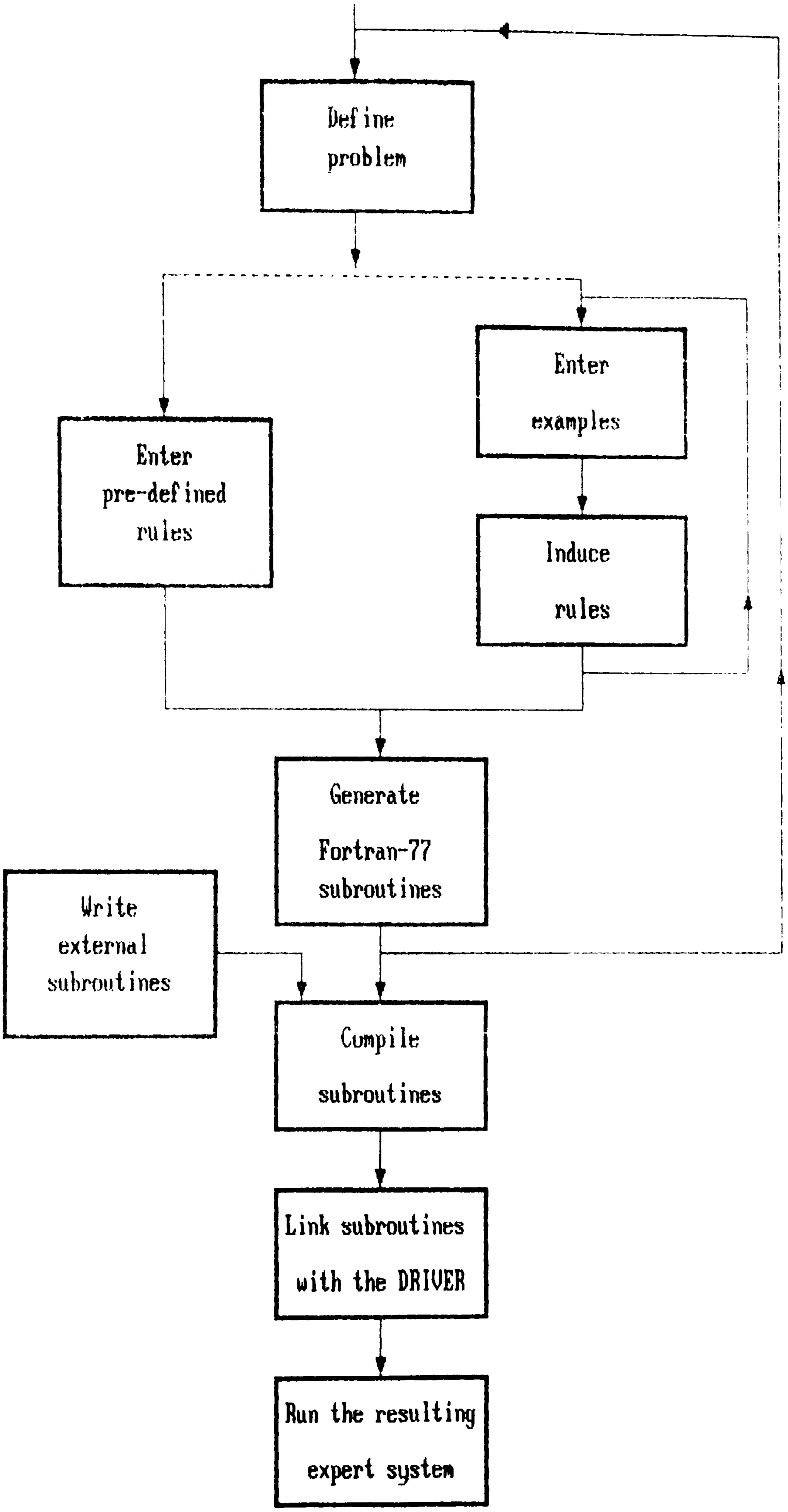

Define problem
Enter pre-defined rules
Enter examples
Induce rules
Generate Fortran-77 subroutines
Write external subroutines
Compile subroutines
Link subroutines with the DRIVER
Run the resulting expert system

Fig. 2.5 Developing expert systems using ExTran

Chapter 3

Modelling and rule based control of a mixing process

3.1 Introduction

To investigate using expert systems in on-line process control, the pilot scale mixing rig in the Control Engineering Laboratory has been taken as an example of an industrial process. Several real-time expert systems, including a rule based control system and various different on-line fault diagnosis systems, have been developed for this process. During the initial developing and testing stage, it would be desirable to develop and test a prototype expert system on the simulation of the process instead of the real one for the following reasons: 1) the simulated process can be run economically, the only demand is computation facilities, while testing on the real process can cost much; 2) the simulated process can be brought to various testing conditions very quickly since it is not running in real time where, in contrast, real industrial processes usually have significantly large time constants and it may take quite a long time to bring a process to a new operating condition; 3) for fault diagnosis systems, any malfunctions can be easily initiated by changing some parameters of the model used in simulation, whereas it may not be convenient to initiate a fault on the real process. From the above considerations, a mathematical model of the mixing process was developed at the initial stage of this research. All the expert systems developed for the mixing process are first tested by simulation. After running satisfactorily on the simulated process, they are then applied to the real process.

The first expert system developed in this research is a rule based on-line control system for the mixing process. It derives control actions from the causal relations among process variables, where the causal relations form a symbolic model of the process. Since the symbolic model captures the causal relations inside a system, for some situations, it can be more understandable than any numerical model. The rule based controller is developed based on the ARTIFACT shell (Francis and Leitch 1985a, b) but is for the multi-input and multi-output case.

The modelling of the mixing process is presented in the next section. The rule based control system is described in Section 3.3, where the causal relations in the mixing process, the control rules, and the performance of the rule based control system are described in detail. The last section contains some concluding remarks.

3.2 Modelling the mixing process

3.2.1 The mixing process

The mixing process is shown in Figure 3.1, where two tanks in cascade and of rectangular cross-section receive hot and cold water input streams. The hot water, at about $80°C$, is supplied from an electrically heated header tank, while the cold water is supplied from the mains. Both streams enter tank 1 where mixing takes place. The contents of tank 1 pass to tank 2 and subsequently out to the pool tank from which they are recycled to the header tank. A number of hand valves can be seen in the mixing process of Figure 3.1. These hand valves are either kept fully open or fully closed during normal operation, as their function is simply to allow different system configurations. For example, if hand valves 1 and 2 are closed and hand valves 3 and 4 are open, then the system becomes a one tank system since only tank 1 can be used. In this research, the two tanks configuration is used and, therefore, hand valves 1 , 2, 3 and 5 are fully open and hand valve 4 is closed. Measurement of level and temperature of the contents of both tanks is available and, hence, it is possible to control level and temperature in either tank.

3.2.2 Model development

A dynamic model of the mixing process can be developed from mass and heat balances in the process. From the mass balance in tank 1, the following equation can be obtained,

$$\frac{d(A_1H_1\rho)}{dt} = \rho(Q_c + Q_h) - \rho Q_{o1} \tag{3.1}$$

where A_1 is the cross-sectional area of tank 1, H_1 is the level in tank 1, ρ is the density of water, Q_c and Q_h are the input cold and hot water flow rates respectively, and Q_{o1} is the output flow rate from tank 1 to tank 2. Eq(3.1) can be simplified to

$$A_1\frac{dH_1}{dt} = Q_c + Q_h - Q_{o1} \tag{3.2}$$

The mass balance in tank 2 can be expressed as

$$\frac{d(A_2H_2\rho)}{dt} = \rho(Q_{o1} - Q_{o2}) \tag{3.3}$$

where A_2 and H_2 are the cross-sectional area and level of tank 2 respectively, and Q_{o2} is the output flow rate from tank 2.

Eq(3.3) can be simplified as

$$A_2\frac{dH_2}{dt} = Q_{o1} - Q_{o2} \tag{3.4}$$

The heat balance in tank 1 can be represented as

$$\frac{d(C\rho A_1H_1T_1)}{dt} = C\rho Q_cT_c + C\rho Q_hT_h - C\rho Q_{o1}T_1 \tag{3.5}$$

where C is the specific heat of water, T_c and T_h are the temperatures of input cold and hot water respectively, and T_1 is the temperature in tank 1. Eq(3.5) can be simplified to

$$A_1H_1\frac{dT_1}{dt} + A_1T_1\frac{dH_1}{dt} = Q_cT_c + Q_hT_h - Q_{o1}T_1 \tag{3.6}$$

Multiply the two sides of Eq(3.2) by T_1 and then substitute it into Eq(3.6), gives

$$\begin{aligned} A_1H_1\frac{dT_1}{dt} &= Q_cT_c + Q_hT_h + Q_{o1}T_1 - Q_cT_1 - Q_hT_1 + Q_{o1}T_1 \\ &= Q_c(T_c - T_1) + Q_h(T_h - T_1) \end{aligned} \tag{3.7}$$

The heat balance in tank 2 gives

$$\frac{d(C\rho A_2H_2T_2)}{dt} = C\rho Q_{o1}T_1 - C\rho Q_{o2}T_2 \tag{3.8}$$

where T_2 is the temperature in tank 2. Eq(3.8) can be simplified to

$$A_2H_2\frac{dT_2}{dt} + A_2T_2\frac{dH_2}{dt} = Q_{o1}T_1 - Q_{o2}T_2 \tag{3.9}$$

Multiplying Eq(3.4) by T_2 and then substitute it into Eq(3.9), gives

$$\begin{aligned} A_2H_2\frac{dT_2}{dt} &= Q_{o1}T_1 - Q_{o2}T_2 - Q_{o1}T_2 + Q_{o2}T_2 \\ &= Q_{o1}(T_1 - T_2) \end{aligned} \tag{3.10}$$

The output flows from the two tanks, Q_{o1} and Q_{o2}, are determined by pressure differences and valve parameters, and can be represented as

$$Q_{o1} = K_1\sqrt{H_1 - H_2} \tag{3.11}$$

$$Q_{o2} = K_2\sqrt{H_2} \tag{3.12}$$

where K_1 and K_2 are the restriction parameters of hand valve 1 and hand valve 2 respectively.

So far, the model of the mixing process is obtained and is listed below.

$$A_1\frac{dH_1}{dt} = Q_c + Q_h - Q_{o1} \tag{3.13}$$

$$A_2\frac{dH_2}{dt} = Q_{o1} - Q_{o2} \tag{3.14}$$

$$A_1H_1\frac{dT_1}{dt} = Q_c(T_c - T_h) + Q_h(T_h - T_1) \tag{3.15}$$

$$A_2H_2\frac{dT_2}{dt} = Q_{o1}(T_1 - T_2) \tag{3.16}$$

$$Q_{o1} = K_1\sqrt{H_1 - H_2} \tag{3.17}$$

$$Q_{o2} = K_2\sqrt{H_2} \tag{3.18}$$

The cross-sectional areas of tank 1 and tank 2 are 17×16.8 cm^2 and 12.4×12.1 cm^2 respectively. The temperature of hot water, T_h, is approximately 80°C, and the temperature of the cold water, T_c, is approximately $20°C$. The other two unknown parameters, K_1 and K_2, are determined from experiments.

3.2.3 Model parameter estimation

The only unknown parameters in the model are the restriction parameters of hand valve 1 and hand valve 2. These parameters are determined from experiments.

An experiment is designed such that the mixing process is operated at its steady state and, therefore, the following equations will hold.

$$Q_{o1} = Q_c + Q_h \tag{3.19}$$

$$Q_{o2} = Q_{o1} \tag{3.20}$$

And from Eq(3.17) and Eq(3.18)

$$Q_c + Q_h = K_1\sqrt{H_1 - H_2} \tag{3.21}$$

$$Q_c + Q_h = K_2\sqrt{H_2} \tag{3.22}$$

In Eq(3.21) and Eq(3.22), H_1 and H_2 are measured variables, Q_c and Q_h are determined by the controlling inputs to the control valves and their values can be calculated from the calibration curves (Ellis *et al* 1986) for the control valves and, therefore, K_1 and K_2 can be calculated.

During the experiment, a set of different values of Q_c and Q_h are applied as inputs to the process, and the corresponding steady state measurements of H_1 and H_2 are recorded. The set of experimental data is listed in Table 3.1. The parameters K_1 and K_2 can be determined from the least squares estimation algorithm(Söderström and Stoica 1989).

For the following model equation

$$y(t) = \varphi^T(t)\theta \tag{3.23}$$

where $y(t)$ is the output, $\varphi^T(t)$ is the input vector, and θ is the parameter vector, if N sets of input and output data are given, then the least squares estimate for θ is

Table 3.1: Experiment data for estimating K_1 and K_2

$H_1(cm)$	$H_2(cm)$	$Q_c + Q_h(cm^3/Sec)$
6.15	2.52	55.6
12.81	5.67	75.3
20.02	8.57	92.5
20.07	10.17	94.3
27.0	15.4	104.0

$$\widehat{\theta} = [\sum_{t=1}^{N} \varphi(t)\varphi^T(t)][\sum_{t=1}^{N} \varphi(t)y(t)] \tag{3.24}$$

The least squares estimate gives

$$K_1 = 29.07(cm^{\frac{5}{2}}/Sec)$$

$$K_2 = 29.46(cm^{\frac{5}{2}}/Sec)$$

3.3 Rule based control of the mixing process

As a first step in this research, a rule based controller for the mixing process is developed. It belongs to the category of "heuristic control" described in Chapter 2 in that the control actions are directly obtained from the expert system. The rule based controller derives control actions from the causal relations between subsystems of the process being controlled. These causal relations form a symbolic model of the process and, for some situations, it can be more understandable than any numerical model. Francis and Leitch (1985a, b) developed an intelligent control system: ARTIFACT, where the process being controlled is similar to the mixing process but is a single-input and single-output (SISO) system. The rule based controller developed here is a development of the ARTIFACT to the multi-input and multi-output case.

3.3.1 Causal relations between subsystems

Level and temperature of tank 2 are to be controlled. The controller is designed based on the causal relations between subsystems and the control actions are inferred from the measurements of both controlled and non-controlled variables.

The mixing process is divided into two subsystems: tank 1 and tank 2. Level and temperature of tank 2 are directly affected by those of tank 1, and the level and temperature of tank 1 are directly affected by the inlet hot and cold flow streams. These causal relations are used to infer control actions.

Based on steady state conditions, an increase in inlet flow will cause the level in tank 1 to increase whereas a decrease in inlet flow will cause the level in tank 1 to decrease. An increase in inlet hot flow or a decrease in inlet cold flow will cause the temperature of tank 1 to increase, while a decrease in inlet hot flow or an increase in inlet cold flow will cause the temperature of tank 1 to decrease. An increase in level and temperature of tank 1 will cause the level and temperature of tank 2 to increase respectively, and a decrease in level and temperature of tank 1 will cause the level and temperature of tank 2 to decrease respectively. These causal relations form a symbolic model of the system.

Based on the symbolic model and the current state of the system, control actions can be inferred. For example, if level 2 is lower than its desired value, then it needs to be increased. If the level in tank 2 is currently not increasing, then the level in tank 1 should be increased. If the level in tank 1 is required to be increased then the inlet flow should be increased.

3.3.2 Control rules

The control rules are in the following form:

$$\text{Goal} + \text{Condition} \Rightarrow \text{Subgoal}$$

where "Goal" is the destination to be achieved, "Condition" is the current state, and the "Subgoal" is the intermediate goal to be achieved under the particular "Condition" in order to achieve "Goal". For example, the following rule:

$$\text{level 2} \uparrow + \text{level 2} - \uparrow \Rightarrow \text{level 1} \uparrow$$

can be interpreted as: "To increase level 2 while level 2 is not increasing, level 1 should be increased."

Since for the level and temperature control loops the symbolic models are identical, they have the same control rules. The full rule sets are listed below.

Rule set 1:

X2 correct + X2 low ⇒ X2 ↑
X2 correct + X2 correct ⇒ X2 steady
X2 correct + X2 high ⇒ X2 ↓

Rule set 2:

X2 ↑ + X2 - ↑ ⇒ X1 ↑
X2 ↑ + [X2 ↑, X1≤A] ⇒ X1 ↑
X2 ↑ + [X2 ↑, X1>A] ⇒ X1 steady
where A is a parameter which is defined later.

Rule set 3:

X2 steady + X2 ↑ ⇒ X1 ↓
X2 steady + X2 steady ⇒ X1 steady
X2 steady + X2 ↓ ⇒ X1 ↑

Rule set 4:

X2 ↓ + X2 - ↓ ⇒ X1 ↓
X2 ↓ + [X2 ↓, X1≥B] ⇒ X1 ↓
X2 ↓ + [X2 ↓, X1<B] ⇒ X1 steady
where B is a parameter which is defined later.

Rule set 5:

X1 ↑ + X1 - ↑ ⇒ Q ↑
X1 ↑ + X1 ↑ ⇒ Q steady

Rule set 6:

X1 steady + X1 ↑ ⇒ Q ↓
X1 steady + X1 steady ⇒ Q steady
X1 steady + X1 ↓ ⇒ Q ↑

Rule set 7:

X1 ↓ + X1 - ↓ ⇒ Q ↓
X1 ↓ + X1 ↓ ⇒ Q steady

When dealing with the level control loop, X1, X2 and Q stand for level in tank 1, level in tank 2, and inlet cold flow respectively. When dealing with the temperature control loop, X1, X2 and Q stand for temperature in tank 1, temperature in tank 2, and inlet hot flow respectively. Within the rule sets the change in Q is proportional to the error between desired value and measured value, with a proportional parameter K. The rules are developed heuristically and the objective is to provide a fast response with low overshoot. This is similar to the ITAE (minimise Integral Time weighted Absolute Error) criterion in optimal control.

These rules are similar to those given by Francis and Leitch (1985a, b) in their ARTIFACT shell. However, rule sets 2 and 4 are different in that the "Condition" parts of the last two rules in each rule set contain extra measurement requirements from tank 1. In rule sets 2 and 4, A and B are determined by the steady state value of X1 corresponding to the setpoint of X2, A is slightly lower than that value whereas B is slightly higher than that value. These two modified rule sets provide a faster response.

3.3.3 Decoupling problem

According to the previous work on controlling this mixing process (Ellis *et al* 1986), the hot inlet flow is used to control temperature and the cold inlet flow is used to control level. Since either hot inlet flow or cold inlet flow can affect both temperature and level, interaction exists between the two control loops. It is necessary to design a decoupling scheme to eliminate the interaction. Heuristic decoupling is used here.

After the control actions for the individual loops have been inferred from the above control rules, they should be modified in order to eliminate interactions. To do this, two situations need to be considered. The first situation is when the hot water flow is changing while the cold flow is being kept steady. Here, in order to eliminate the effect of changing hot flow on the level control loop, the total amount of inlet water flow should be unchanged. That is:

$$\Delta Q_c + \Delta Q_h = 0 \qquad (3.25)$$

Therefore

$$\Delta Q_c = -\Delta Q_h \qquad (3.26)$$

So, in this situation, the final control action is that the cold water inlet flow should be changed by $-\Delta Q_h$.

The other situation is when the hot water inlet flow is being kept unchanged while the cold water inlet flow is changing. Here, in order to eliminate the effect of changing the cold water inlet flow on the temperature control loop, the total input heat should be unchanged. That is:

$$\Delta Q_c \rho C(T_c - T_1) + \Delta Q_h \rho C(T_h - T) = 0 \tag{3.27}$$

Therefore

$$\Delta Q_h = (T_1 - T_c)Q_c/(T_h - T_1) \tag{3.28}$$

So, in this situation, the final control action is that the hot water inlet flow should be changed by $(T_1 - T_c)\Delta Q_c/(T_h - T_1)$.

In Equations (3.25) – (3.28), ΔQ_c is the change in cold inlet flow, ΔQ_h is the change in hot inlet flow, ρ and C are the density and specific heat of the inlet water respectively, T_1 is the temperature of tank 1, T_c is the temperature of inlet cold flow, and T_h is the temperature of the inlet hot flow.

3.3.4 Knowledge base and inference engine

The knowledge base consists of the control rules, the decoupling rules, and some general knowledge about the control system, such as control valve saturation. When a control valve saturates, its output will not change, and this situation should be dealt with differently from that discussed above. The inference engine simply performs forward chaining (Johnson and Keravnou 1984, Jackson 1986). Control rules are chained together by the "Goal" and "Subgoal" parts of each rule. Initially, the "Goal" is assigned a value: X2 correct. The rule satisfies the current condition and the value of the "Goal" is employed and the value of the "Sub-goal" is renewed. The inference engine continuously performs this procedure until the value of the "Goal" refers to inlet flow. Then using decoupling rules to eliminate interactions, control actions are obtained.

3.3.5 Performance of the rule based controller

The rule based controller has been implemented using a BASIC program running on a BBC microcomputer. Its performance is very satisfactory as can be seen from Figures 3.2 and 3.3, where the performance of the rule based controller has been compared with that of a conventional decoupling PI controller designed by Ellis *et al* (1986), for step changes in temperature and level respectively. It can be seen that the performance of the rule based controller matches, qualitatively, that of the decoupling PI controller. The response of the rule based controller has low overshoot and undershoot and the interaction between the two loops is also very slight. From Figure 3.2 it can be seen that the rule based controller has been attempting to achieve the objective "fast response with low overshoot".

The tuning of the rule based controller is done on-line by adjusting the parameters K, A and B, and is relatively easy. It has been found that the controller is not very sensitive to change of tuning parameters. This suggests that the properties of a rule based controller is largely determined by its rules. The role of controller parameters is less crucial in rule based controllers than that in conventional controllers.

3.4 Conclusions

A mathematical model for the mixing process has been developed in this chapter. The model is used to test several prototype real-time expert systems developed for the mixing process. By such means, real-time expert systems can be developed quickly.

The rule based controller described in this chapter has been observed to perform satisfactorily. This suggests that it could be an alternative for conventional controllers in cases where numerical models for the controlled processes are not available or are difficult to obtain. The properties of a rule based controller are mainly determined by its rules, and it is observed that the rule based controller is not very sensitive to the changes in its parameters. This may demonstrate the robustness of rule based controllers.

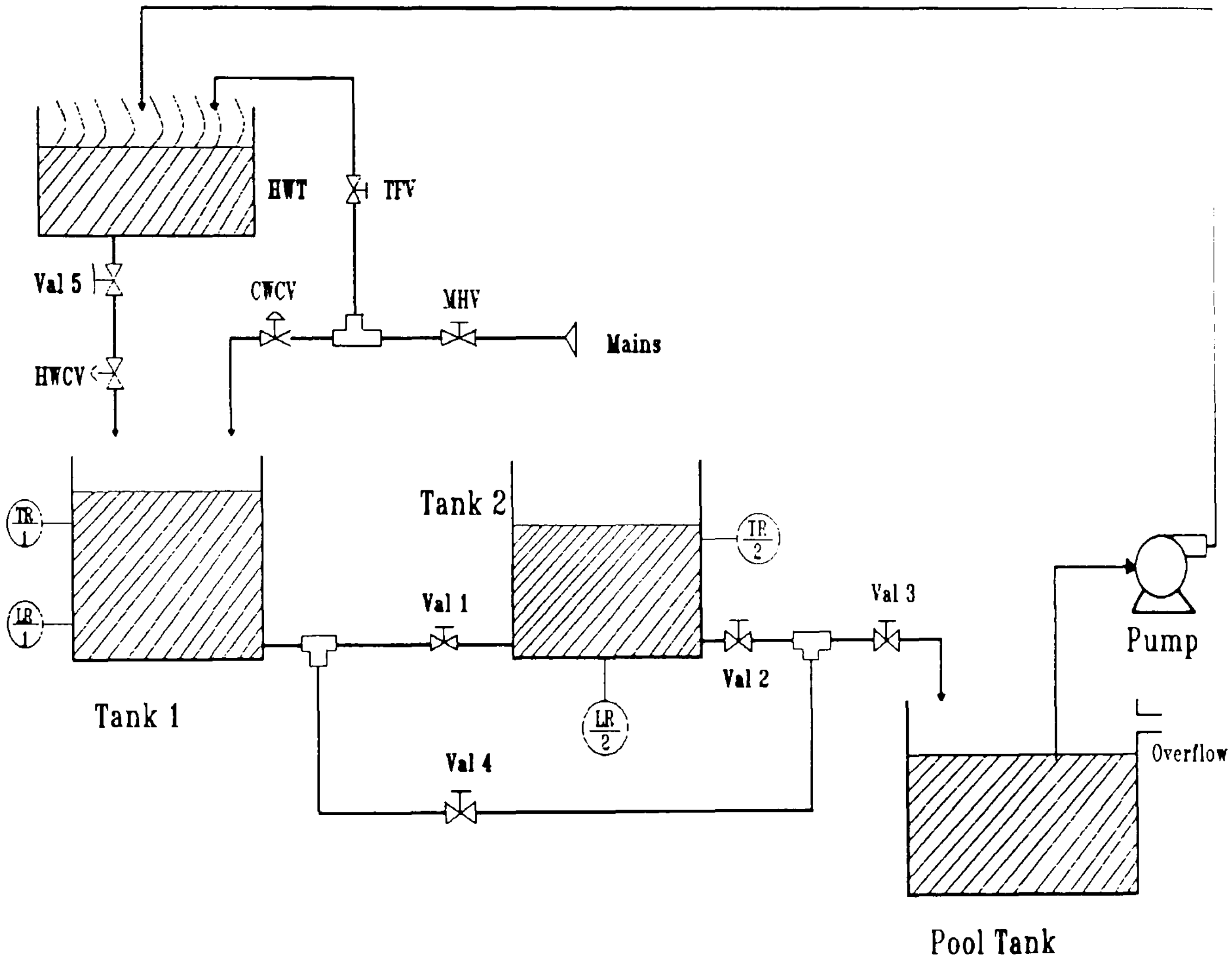


Fig. 3.1 The mixing process

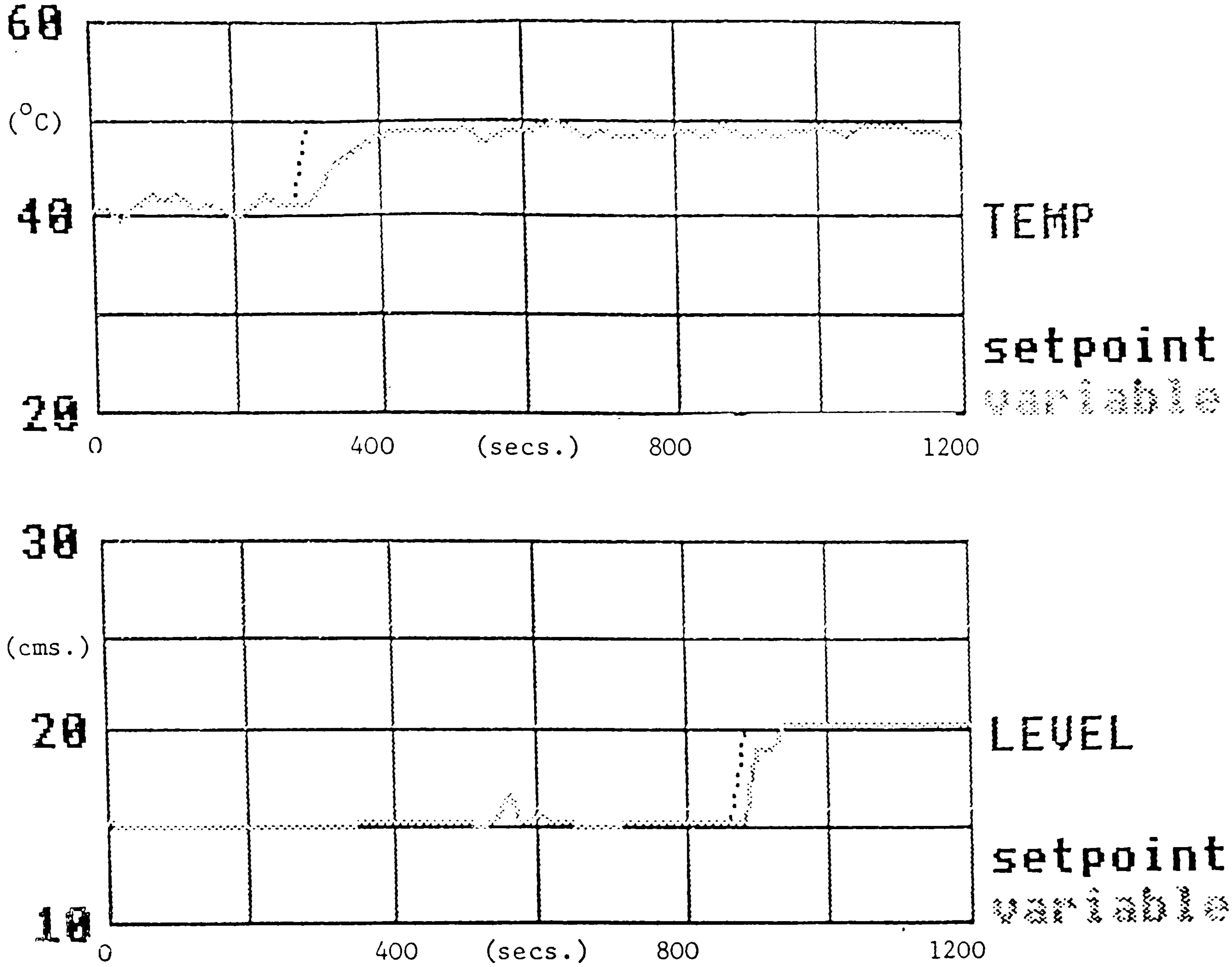


Figure 3.2 Response of the rule based controller

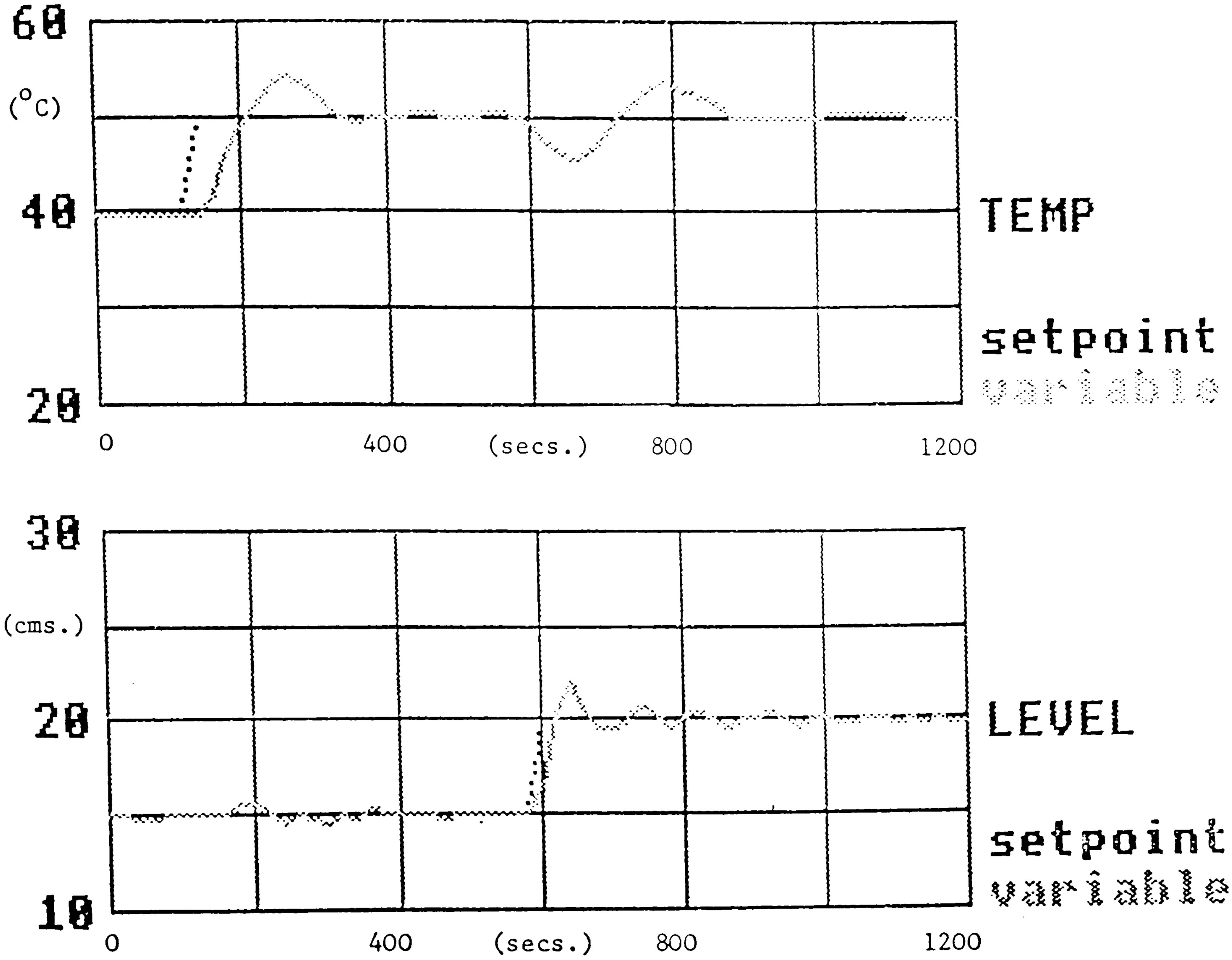


Figure 3.3 Response of the decoupling PI controller

Chapter 4

Process fault diagnosis from knowledge on system structures and component functions

4.1 Introduction

Process equipments are subject to failures during operation. Failures may cause poor product quality, reduce production efficiency, damage equipment, lead to plant shutdowns, or even result in a hazardous condition. Therefore, it is important to continuously monitor the process in order to detect and diagnose faults promptly. This task is traditionally carried out by process operators. As the process becomes more and more complex, the number of measurements and alarms increase and may cause cognitive overload to process operators (Paterson, Sachs, and Turner 1985). In such situations, the process operator may not provide the correct diagnosis in limited time and, furthermore, the reliability of an operator is likely to suffer when forced to make quick judgment or forced to depend upon operating and safety manuals which may not be written in a clear or concise fashion. Therefore, automated fault diagnosis is required, the importance of which increases as the processes become more and more complex. Knowledge based systems show a great potential in this field.

As described in Chapter 2, knowledge based diagnosis systems can be generally divided into shallow knowledge based and deep knowledge based approaches. In the shallow knowledge based approach, the diagnostic knowledge used is the process operators' experience, which directly reflects the relations between symptoms and faults, and is organised as cause consequence rules as used, for example, in MYCIN (Johnson and Keravnou 1984, Harmon and King 1985). Although these heuristic rules possess real-time efficiency, they lack process generality and they tend to fail under novel circumstances. Recently reported diagnosis systems for industrial processes often use the deep knowledge based approach or use a combined approach where deep knowledge plays a dominant role. The advantages of deep knowledge based approaches are that they can provide reliable diagnosis for infrequently occurred faults, and some of the deep knowledge is general in nature and can be used in the development of diagnosis systems for other processes.

One of the deep knowledge based approaches is causal search (Moor and Kramer 1986). In this approach, the diagnosis system will try to explore the causal path from the observed abnormalities to their causes and, therefore, locate any associated faults. To improve efficiency, the process under consideration can be decomposed into several subsystems. Finch and Kramer (1988) propose a diagnostic method based on functional decomposition of an industrial process. In their approach, the process under consideration is decomposed into several subsystems according to their functions, then diagnosis is performed by identifying the source system, which is the subsystem where the fault occurs, and then locating this fault in the source system. Steels (1989) investigates a similar approach where the function of the system being diagnosed is hierachically decomposed.

In this research, a diagnosis approach based on structural decomposition is investigated. Since structural decomposition corresponds to plant topology, it may be easier to implement. The relations between subsystems, the relations among measured variables inside a subsystem or in two related subsystems, and the relations between faults and measurements in a subsystem are represented by several matrices. Diagnostic rules can be developed from this knowledge. Benefits of rule-based format are that the rules can be evaluated efficiently and can be combined with other rules pertaining to plant operations. When abnormalities occur in a process subsystem under consideration, they are traced through other subsystems affecting this subsystem until a source subsystem is located. Once a source subsystem is located, the diagnosis system will identify the malfunction in the source subsystem.

A general structure for the on-line diagnosis system is described in the next section, all the diagnosis systems developed in this research are based on this structure. Section 4.3 describes how to formulating diagnostic rules from knowledge on structures and functions. Section 4.4 describes the development of an on-line fault diagnosis system for the pilot scale mixing process. A fault diagnosis system for a simulated continuously stirred tank reactor (CSTR) process is described in Section 4.5, where the modelling of the CSTR is also presented. The last section contains some conclusions.

4.2 General structure of the on-line diagnosis system

The on-line fault diagnosis system resides in the supervisory layer of a hierachical control system shown in Figure 4.1, where a process is controlled by a controller in the control layer, and this control layer also communicates with the supervisory layer. There are many supervisory functions, such as determining the optimal setpoints and monitoring the condition of the control system, and fault detection and diagnosis is one of them. The controller in the control layer simply performs regulation tasks, and the sampling interval of the controller is T. The communication interval between the supervisory layer and the control layer is nT, where n is a positive adjustable integer. This communication interval can be set longer for normal operating conditions and shorter for abnormal conditions.

The diagnosis system contains two parts: abnormal behaviour detection and fault diagnosis. During normal operation, the supervisory layer receives data from the regulatory layer at the interval nT. The fault diagnosis system examines the data to find out if it is abnormal or not. If it is abnormal, then the communication interval between the supervisory and the regulatory layers is reduced. The diagnosis system then swiftly collects several additional sets of data, and examines if the detected abnormalities are present in the majority of those sets of data. Suppose N sets of data are collected, then

$$AB(m_i) \Leftarrow N_a^{m_i} \geq N_t$$

which states that abnormal behaviour in m_i is detected if the number of sets of data where m_i is abnormal, $N_a^{m_i}$, is greater than or equal to its threshold value N_t. Once abnormal behaviour of the process is detected, the diagnosis system begins to locate the associated fault. By swiftly collecting several additional sets of data, the effect of measurement noise can be eliminated to some extent.

Abnormal behaviour detection can be performed by checking certain measurements against their constraint values, checking the range of change of some measurements, and examining if some constraints, such as those imposed by energy and mass balance, are violated.

The diagnosis system diagnoses faults based on on-line information, which contains on-line measurements and controller outputs. It is suggested that performing tests on the diagnosed system could help the generation of hypotheses and the discrimination of candidate faults (Milne 1987). This is used in the diagnosis of electronic and digital circuits (Davis 1983, 1984). In the method proposed by Yamada and Motoda (1983), tests using redundant components are used to discriminate suspects. In general, for control systems without redundant components, performing tests may disturb the process. To avoid this, the diagnosis systems developed in this research diagnose faults from the available on-line information and do not perform any intrusive tests on the process. Therefore, employing such diagnosis systems will not have any side effects on the controlled process. Even though it may provide a wrong diagnosis or miss a fault, the diagnosis system will never affect the controlled process. Since most of the reported on-line fault diagnosis systems are tested on pilot scale processes or simulated processes, the above consideration would be important for developing on-line fault diagnosis systems for real industrial process.

One feature of a real-time diagnosis system is that it has a dynamic knowledge base. The factual knowledge is dynamically changing. During diagnosis, not only the current on-line information but a history of the process states is needed. The diagnosis system will maintain a memory of a short history of the monitored process and this memory is dynamically renewed by on-line information.

The diagnosis system also has a "repair flag", which will be set automatically after a diagnosis to disable the diagnosis system. After repairing, the process operators can reset this flag to enable the fault diagnosis system. During setpoint changes, this flag is also set automatically for a period to allow the process to settle down. Process operators can set or reset this flag as is required.

4.3 Formulating diagnostic rules from knowledge on system structures and component functions

4.3.1 Description of system structures

In order to narrow the diagnosis focus the process under consideration is structurally decomposed into several subsystems, where the structural decomposition corresponds to the plant topology. The process can be briefly represented by a diagnosis graph, which contains nodes and directed arcs. Each node represents a subsystem and the arcs represent interactions between subsystems. The diagnosis graph is similar to the Signed Directed Graph (SDG) (Iri *et al* 1979). In a SDG, each node represents a process variable, whereas in the diagnosis graph each node represents a subsystem. An example diagnosis graph is shown in Figure 4.2, where the process is divided into four subsystems. Directed arcs in Figure 4.2 show that subsystem S_1 can affect subsystem S_2, subsystems S_2 and S_3 can affect each other, and subsystem S_4 can affect both subsystems S_2 and S_3.

The interactions among subsystems can be represented by the Connection Matrix, C. If the process is decomposed into n subsystems, then C is an $n \times n$ matrix. The element of C, C_{ij}, is defined as follows:

$$C_{ij} = \begin{cases} 1, & \text{if subsystem } S_i \text{ can directly affect subsystem } S_j, \\ 0, & \text{otherwise.} \end{cases}$$

The diagonal elements of the Connection Matrix are all ones since a subsystem can affect itself.

The state of a subsystem is described by its measurements and a subsystem is abnormal if one of its measurements is abnormal, that is

$$AB(S_i) \Leftarrow \exists k,\ k \in \overline{1, m_i},\ AB(m_{ik})$$

which states that if there exists a measurement, m_{ik}, which is abnormal, in subsystem S_i, then subsystem S_i is abnormal. In the above expression, AB is a predicate meaning abnormal, m_i is the total number of measurements in S_i, m_{ik} is the kth measurement in S_i.

In the connection matrix, if $C_{ij} = 1$, then subsystem S_i can affect subsystem S_j. This means that one or some of the process variables in S_i can affect those in S_j. The connection matrix only provides a rough description on the relations among subsystems. A refined description can be given by the Measurement Causal Matrix, CM_{ij}. If there are n measurements in S_i and m measurements in S_j, then the Measurement Causal Matrix between S_i and S_j, CM_{ij}, is an $n \times m$ matrix. The element of CM_{ij}, CM_{ij}^{kl}, is determined as

$$CM_{ij}^{kl} = \begin{cases} 1, & \text{if the } kth \text{ measured variable in } S_i \text{ can directly} \\ & \text{affect the } lth \text{ measured variable in } S_j, \\ 0, & \text{otherwise.} \end{cases}$$

There also exist causal relations between measured variables within a subsystem. These relations are represented by the Self-Causal Matrix. If there are n measurements in subsystem S_i, then the Self-Causal Matrix for subsystem S_i, CS_i, is an $n \times n$ matrix. The element of CS_i is determined as follows:

$$CS_i^{kl} = \begin{cases} 1, & \text{if the } kth \text{ measured variable in } S_i \text{ can directly} \\ & \text{affect the } lth \text{ measured variable in } S_i, \\ 0, & \text{otherwise.} \end{cases}$$

The diagonal elements of the Self-Causal Matrix are all ones since a measurement can affect itself.

To locate faults in a subsystem, the relations between faults and measurements in that subsystem should be taken into account. These relations can be represented by the Fault-Measurement Matrix. If there are n possible malfunctions and m measurements in subsystem S_i, then the Fault-Measurement Matrix for subsystem S_i, FM_i, is an $n \times m$ matrix. The element of FM_i, FM_i^{kl}, is determined as

$$FM_i^{kl} = \begin{cases} 1, & \text{if the } kth \text{ malfunction in } S_i \text{ can directly} \\ & \text{affect the } lth \text{ measurement in } S_i, \\ 0, & \text{otherwise.} \end{cases}$$

The diagnosis graph and the above defined matrices give a description of the process being diagnosed. Diagnostic rules can be generated from this description.

4.3.2 Fault diagnosis based on knowledge of system structure and component functions

With the above described structural decomposition and knowledge on system structures and component functions, fault diagnosis can be performed in the following two step procedure: source subsystem identification and fault location in the source subsystem. Because of the dependence between subsystems, the effect of a fault can propagate through connected subsystems and, therefore, a fault can not only affect measurements of the subsystem where it occurs but also affect measurements of other connected subsystems. When abnormal behaviour is detected, the first step in diagnosis is to identify the source subsystem by causally tracing the observed abnormalities. To facilitate diagnosis, the "single-failure assumption", which is used in most fault diagnosis systems (Davis 1983, 1984, Finch and Kramer 1988), is adopted here. This assumption is also practical since the probability of simultaneous occurrence of two or more independent faults is usually negligible. Suppose that two independent faults F_1 and F_2 occur with probabilities P_1 and P_2 respectively, then the probability of a simultaneous occurrence of F_1 and F_2 is P_1P_2, which in general would be too small to take into account.

Suppose that the *jth* measurement in the *ith* subsystem is abnormal, that is $AB(m_{ij})$, then a search is conducted to causally search any measured variables in S_i which can cause the observed abnormality in m_{ij} and, if such a variable exists, then it is activated, which means that it is responsible for the observed abnormality. This search is guided by the Self-Causal Matrix of subsystem S_i. Similar searches are also conducted to find further causes in S_i for the activated variable. Suppose that the finally activated variable in S_i is m_{ik}, then a search is conducted to find all the subsystems that are connected to S_i. These subsystems form the set

$$\{\forall j,\ S_j,\ C_{ij} = 1,\ j \neq i\} \tag{4.1}$$

Next, pick a subsystem from the above set, for example S_j, and conduct a search to find all the measured variables in S_j that could affect m_{ik}. These measurements form a set

$$\{\forall l,\ m_{jl},\ CM_{ji}^{ik} = 1\} \tag{4.2}$$

The above set can be refined by examining the deviations of these measurements and their causal relations with m_{ik} and only the measurements which could result in the observed deviations in m_{ik} are retained. If the refined set is empty, then try other subsystems in Set(4.1), and if the resulting sets are all empty, then subsystem S_i is a source subsystem. If there exists a refined set which is not empty, then pick a measurement from the set as an activated variable and conduct further searches similar as above.

Once a source subsystem is identified, the remaining task is to locate the fault in the source subsystem. Suppose that S_k is a source subsystem and m_{kl} is the finally activated measurement, then a candidate fault set is formulated as

$$\{\forall i,\ F_{ki},\ FM_k^{il} = 1\} \tag{4.3}$$

where F_{ki} is the ith malfunction in subsystem S_k. The above set can be refined by examining the patterns of deviations of measurement m_{kl} and its causal relations with these candidate faults and only the malfunctions which can lead to the observed deviations in m_{kl} are retained. Certain process specific heuristic rules can be used in this stage. Based on the above consideration, diagnostic rules can be formulated.

A benefit of the rule based format is that the diagnostic rules can be augmented by any available heuristic knowledge about a particular process. The above described procedure will be demonstrated in the development of diagnostic rules for a pilot scale mixing process and a simulated CSTR system in the next two sections.

4.4 On-line fault diagnosis of the mixing process

The first diagnosis system developed in this research is the on-line fault diagnosis system for the mixing process. It is initially developed using Fortran (Zhang, Roberts, and Ellis 1988). After the Control Engineering Centre has purchased an expert system shell: **ExTran**, the diagnosis system is redeveloped using **ExTran**.

4.4.1 Abnormal behaviour detection

Constraint values have been assigned to every measured variable and, if the measurement exceeds its constraint value, it is considered to represent abnormal behaviour. For the controlled variables, in addition to the constraint values, error tolerances have been set which, together with the changing direction of the controlled variables, can be used to detect abnormal behaviour, and thus the abnormal behaviour can be found quickly. Some general knowledge about the system performance is also used to detect abnormality. For example, in the steady state, the level in tank 1 cannot be lower than that in tank 2 and the temperatures of the contents of the two tanks are roughly the same. Any abrupt changes in sensor readings are also considered as abnormal.

After receiving the data, the diagnosis system examines them to see whether they are normal or not. A memory of a short history of the process is kept which is used to determine any abnormal behaviour and is also used for diagnosis. Under normal conditions, the memory is renewed by newly received data, in that the new data replaces the earlier data in the memory. When an abnormal condition is detected the earlier data in memory is retained and several sets of additional data are swiftly collected by increasing the speed of communication between the two layers. The resulting information is used to confirm the abnormality detection and is also used for diagnosis. After this data has been collected the communication speed is set to normal again. If the majority of the collected data declare the same abnormality, then the abnormal behaviour is confirmed, otherwise, the behaviour is still considered to be normal. By this means the effects of noise on the measurements can be considerably reduced.

In this system, the controller sampling time is 4 seconds and, under normal conditions, the local controller sends data to the supervisory layer every 20 seconds, that is every 5 samples. Four sets of data are kept as a short history. If the received data indicates an abnormal condition, the controller then sends data to the supervisory layer every 4 seconds, that is at every sample, until 4 more sets of data have been transmitted.

4.4.2 Formulation of diagnostic rules

The mixing process is divided into two subsystems. The first subsystem includes the following components: controller, hot and cold water control valves, tank 1 and the associated sensors. Components in the second subsystem are hand valves 1 and 2, tank 2 and the associated sensors. The diagnosis graph corresponding to this decomposition is shown in Figure 4.3, from which it can be seen that the two subsystems can affect each other. The level and temperature in the second subsystem are affected by those in the first subsystem. The controller outputs in the first subsystem are affected by the controlled variables in the second subsystem. The Connection Matrix for the mixing process is

$$C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{4.4}$$

The on-line information in the first subsystem includes level and temperature measurements and controller outputs to the cold and hot water control valves. The Self-Causal Matrix for the first subsystem is

$$CS_1 = \begin{array}{c} \\ H_1 \\ T_1 \\ Q_c \\ Q_h \end{array} \begin{array}{c} \begin{array}{cccc} H_1 & T_1 & Q_c & Q_h \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \tag{4.5}$$

The labels on the top and the left of the matrix, H_1, T_1, Q_c, and Q_h, are level and temperature measurements in tank 1, and controller outputs to cold water and hot water control valves respectively. In the mixing process, either Q_c or Q_h can affect both H_1 and T_1, however, since Q_c is used to control level and Q_h is used to control temperature, the effect of Q_c on T_1 and the effect of Q_h on H_1 can be eliminated by the feedback control loops and, therefore, Eq(4.5) indicates that Q_c cannot affect T_1 and Q_h cannot affect H_1.

The on-line information for the second subsystem is the level and temperature measurements in tank 2. The Self-Causal Matrix for the second subsystem is

$$CS_2 = \begin{array}{c} \\ H_2 \\ T_2 \end{array} \begin{array}{c} \begin{array}{cc} H_2 & T_2 \end{array} \\ \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right] \end{array} \tag{4.6}$$

The labels on the top and the left of the matrix, H_2 and T_2, are the level and temperature measurements in tank 2 respectively.

The Measurement Causal Matrix from subsystem 1 to subsystem 2 is

$$CM_{12} = \begin{array}{c} \\ H_1 \\ T_1 \\ Q_c \\ Q_h \end{array} \begin{array}{c} \begin{array}{cc} H_2 & T_2 \end{array} \\ \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array}\right] \end{array} \tag{4.7}$$

The above equation indicates that Q_c and Q_h cannot affect H_2 and T_2, which is due to the fact that Q_c and Q_h cannot directly affect H_2 and T_2 since their influence on H_2 and T_2 is exerted via measured variables H_1 and T_1 respectively.

The Measurement Causal Matrix from subsystem 2 to subsystem 1 is

$$CM_{21} = \begin{array}{c} \\ H_2 \\ T_2 \end{array} \begin{array}{c} \begin{array}{cccc} H_1 & T_1 & Q_c & Q_h \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{4.8}$$

The controller used here is a multivariable controller and, therefore, either H_2 or T_2 can affect both Q_c and Q_h. However, Eq(4.8) indicates that H_2 can only affect Q_c, and T_2 can only affect Q_h. This is due to the fact that Q_c and Q_h are dominantly affected by H_2 and T_2 respectively.

The faults that may occur in the first subsystem are considered to be: controller failure, control valve failures, and sensor failures. The Fault Measurement Matrix for the first subsystem is

$$FM_1 = \begin{array}{c} \\ C \\ H \\ ST1 \\ SL1 \\ CO \end{array} \begin{array}{c} \begin{array}{cccc} H_1 & T_1 & Q_c & Q_h \end{array} \\ \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right] \end{array} \tag{4.9}$$

The labels on the left of the matrix, $C, H, ST1, SL1$, and CO, stand for cold and hot water control valve failures, temperature and level sensor failures, and controller failure respectively. The above equation indicates that cold water control valve failure cannot affect T_1 and hot water control valve failure cannot affect H_1. This is due to the fact that the effects of cold water control valve failure on T_1 and hot water control valve failure on H_1 are compensated by feedback control loops.

The faults that may occur in the second subsystem are considered to be: blockage of hand valves 1 and 2, and sensor failures. The Fault Measurement Matrix for the second subsystem is

$$FM_2 = \begin{array}{c} \\ V1 \\ V2 \\ ST2 \\ SL2 \end{array} \begin{array}{c} \begin{array}{cc} H_2 & T_2 \end{array} \\ \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{array}\right] \end{array} \tag{4.10}$$

The labels on the left of the matrix, $V1, V2, ST2$, and $SL2$, stand for blockages of hand valve 1 and 2, and failures of temperature sensor and level sensor in tank 2 respectively.

Based on the above described system structures, diagnostic rules can be developed. The rules are developed using the **ExTran** expert system shell (Razzak, Hassan, and Ahmad, 1986) and the diagnosis system is defined by a main problem and six subproblems. The main problem classifies the observed abnormalities, and different abnormalities are treated by different subproblems. Since there are only four measurements, correspondingly, there are four kinds of abnormalities. The outcomes of the main problem are four different subproblems each corresponding to a type of abnormality. The rule files for the main problem and other six subproblems are shown in Figure 4.4, and the definitions of the attribute values are given in Table 4.1. The values of these attributes are evaluated by external Fortran subroutines from on-line measurements.

It can be seen from Table 4.1 that the information handled by the diagnosis system is in qualitative form which is converted from on-line quantitative information. The conversion is usually performed by comparing on-line information with certain threshold values. The threshold values used in fault detection and fault diagnosis will affect the performance of the diagnosis system and should be set properly. Small threshold values could make the diagnosis system sensitive to process disturbances and measurement noise, and may result in spurious diagnosis. Large threshold values may miss a diagnosis. During the current studies, it is found that the proper setting of these threshold values used in fault detection can remarkably reduce spurious diagnoses. These parameters are set based on previous operational experience of the process and should be set reasonably large so that any fluctuations in measurements caused by disturbances will not trigger the diagnosis system.

Subproblem TEMP2 will be invoked if abnormalities are present in the measurement of T_2. The rules for this subproblem are developed from the following considerations. There are two situations when T_2 is abnormal, one is that T_2 is lower than its set point, and another one is that T_2 is higher than its set point. Consider the first situation. Following the procedure described in the previous section, a search is conducted to find if there are any measured variables in the second subsystem which can affect T_2. From Eq(4.6), it can be seen that no such variables exist. Eq(4.4) indicates that the first subsystem can affect the second subsystem and, furthermore, Eq(4.7) shows that only T_1 in the first subsystem can affect T_2 in the second subsystem. Then T_1 should be examined to locate the source subsystem. If T_1 is decreasing, then it is activated, otherwise the second subsystem is a source subsystem. If T_1 is activated, then from Eq(4.5) it can be seen that the controller output to the hot water control valve, Q_h, can affect T_1. If Q_h is decreasing, then it is responsible for the decrease in T_1 and is activated. In this case, the search for the source subsystem is terminated since both subsystems have been explored and no further variables can be activated, and the task is to locate a fault in the first subsystem. Eq(4.9) suggests that only controller failure can affect Q_h and, therefore, the conclusion is controller failure. If Q_h is not decreasing, then from Eq(4.9) the candidate failures would be hot water control valve failure and sensor T_1 failure. Sensor T_1 failure can be ruled out by the single failure assumption since it cannot explain the abnormality in T_2. If T_1 is not decreasing, then the second subsystem would be a source subsystem. In this case, Eq(4.10) suggests that only sensor T_2 failure can affect T_2, and then the subproblem SENST2, which contains several heuristic rules for diagnosing sensor failure, is used to further confirm that sensor T_2 has failed. The rules for the second situation where T_2 is not lower than

Table 4.1: Definitions of attributes in the diagnosis system for the mixing process

Attributes	Definitions
ABS2	Subsystem 2 is abnormal
ABT1	Temp. 1 is abnormal
ABT2	Temp. 2 is abnormal
T2LTSP	Temp. 2 is lower than its setpoint
T1DCR	Temp. 1 is decreasing
T1INC	Temp. 1 is increasing
H1DCR	Level 1 is decreasing
H1INC	Level 1 is increasing
QCDCR	Q_c is decreasing
QCINC	Q_c is increasing
QHDCR	Q_h is decreasing
QHINC	Q_h is increasing
H2LTSP	Level 2 is lower than its setpoint
H2COND	Level 2 is continuously decreasing
H2CONI	Level 2 is continuously increasing
T2SC	There are abrupt changes in temp. 2
T1SC	There are abrupt changes in temp. 1
H2SC	There are abrupt changes in level 2
H2SC	There are abrupt changes in level 1
T1IT2S	Temp. 1 increasing & temp. 2 steady
T1DT2S	Temp. 1 decreasing & temp. 2 steady
H1IH2S	Level 1 increasing & level 2 steady
H1DH2S	Level 1 decreasing & level 2 steady
T2NRSP	Temp. 2 is near its setpoint
H2NRSP	Level 2 is near its setpoint
DTH	Deference between temp. 1 & 2 is high
DHH	Deference between level 1 & 2 is high

its set point are developed similarly as above.

If an abnormality is present in H_2, then subproblem LEVEL2 will be used. The development of rules for this subproblem is based on the following considerations. From Eq(4.4) it can be seen that S_1 can affect S_2 and, furthermore, Eq(4.7) shows that only H_1 in S_1 can affect H_2. Consider the situation where H_2 is lower than its set point. If H_1 is decreasing, then S_1 will be a source subsystem and H_1 is activated, otherwise, S_2 is a source subsystem. If H_1 is activated, from Eq(4.5) Q_c can affect H_1, and if Q_c is decreasing then Q_c is activated, and Eq(4.9) suggests that the only candidate fault is controller failure. If Q_c is not decreasing, then Eq(4.9) shows that two failures: cold water control vale failure and level sensor 1 failure, could affect H_1. The single failure assumption rules out the failure of level sensor 1 since it cannot account for the abnormality in H_2. If S_2 is a source subsystem,then Eq(4.10) suggests three candidate failures: level sensor 2 failure, blockages of hand valves 1 and 2. The blockage of hand valve 2 is ruled out since it could not cause H_2 to decrease. The remaining two candidates are discriminated by the heuristic that if hand valve 1 is blocked H_2 will decrease continuously. Therefore, if H_2 is decreasing continuously, then hand valve 1 is blocked, otherwise, sensor H_2 may fail which is further discriminated by the subproblem SENSH2. The derivation of diagnostic rules for the situation that H_2 is higher than its set point is similar to the above considerations.

The subproblems SENST2 and SENSH2 are used to discriminate failures of sensor T_2 and sensor H_2 respectively. The rules for the two subproblems are similar and contain some heuristics about sensor failure. The first heuristic is that abrupt changes in sensor readings indicate sensor failure. Since the measured variables in the mixing process have large time constants, especially the levels, they cannot change abruptly. Another heuristic is that if T_1 (or H_1) is changing in the direction to move T_2 (or H_2) to its set point, but T_2 (or H_2) does not change, then sensor T_2 or (H_2) failure is indicated. The subproblems SENST1 and SENSH1 are used to discriminate failures of sensor T_1 and sensor H_1 respectively. The rules for the two subproblems are similar and the first rule is the same as that in SENST2 and SENSH2. The second heuristic is that if H_2 (or T_2) is near to its set point and the difference between H_1 and H_2 (or T_1 and T_2) is high, then sensor H_1 (or T_1) failure is indicated.

4.4.3 Performance of the fault diagnosis system

The fault diagnosis system has been successfully applied to the mixing process. In the mixing process, the possible faults that can occur are: controller failure, sensor failure, hot and cold water control valves failure, hand valve 1 blocked, and hand valve 2 blocked. During the experiments, these faults have been separately initiated, and they were diagnosed quite successfully. The failures of control valves are initiated by turning off the hand valves in series with the control valves totally or partially, the blockages of hand valves are initiated by turning them off totally or partially, and sensor failures are initiated by disconnecting them from the A/D (analogue to digital) converter.

The on-line measurements covering the event where hot water control valve failure was initiated and diagnosed are shown in Figure 4.5. The failure was initiated by turning off the hand valve in series with the hot water control valve (see Figure 3.1). The fault diagnosis system detected abnormality in measurements after 356 seconds then, as indicated in Figure 4.5, the diagnosis system swiftly collected another four more sets of data to confirm abnormal behaviour detection, after which the diagnosis system begins to diagnose the fault. The reasoning procedure of the fault diagnosis system is recorded in a file by **ExTran** and is shown in Figure 4.6, which indicates that the correct diagnosis is presented. After diagnosis, the fault is removed and all measurements settle down to their steady state values as indicated in Figure 4.5. Figure 4.5 also indicates that abnormalities in measurements were detected after 240 seconds, but after collecting another four more sets of data, abnormal behaviour was not detected. This could have resulted from a disturbance in the process.

Table 4.2 shows the result of the experiments in which every fault was initiated five times. It can be seen that the performance of the fault diagnosis system is very satisfactory. Since the detection of abnormal behaviour is based on steady state measurements, when changing setpoints, it is important to wait for sufficient time to allow transient effects to decay before initiating the fault diagnosis system.

Most of the existing fault diagnosis systems are based on a single failure assumption (Davis 1983, 1984, Scarl, Jamieson, and Delaune 1987, Finch and Kramer 1988). The fault diagnosis system presented here is also designed for diagnosis of a single fault. After one fault has been diagnosed, any further faults will not be diagnosed. During some experiments, several faults were initiated simultaneously and, in most of the cases, one of the initiated faults can be diagnosed. Table 4.3 shows the performance of the diagnosis system when several faults were initiated

Table 4.2: Performance under a single failure

fault initiated	no. of successful diagnoses
temp. sensor 1 fail	5
temp. sensor 2 fail	5
level sensor 1 fail	5
level sensor 2 fail	5
hand valve 1 blocked	5
hand valve 2 blocked	5
hot water control valve fail	5
cold water control valve fail	5

simultaneously.

4.5 Fault diagnosis of a CSTR system

Following similar procedures as described in the previous section, a fault diagnosis system is developed for a simulated CSTR system, similar to that used by Kramer and co-workers (Kramer and Palowitch 1987, Finch and Kramer 1988, Oyeleye and Kramer 1988, Kramer and Finch 1989). The CSTR system is shown in Figure 4.7, where a hypothetical exothermic reaction takes place in the reactor vessel, cooled by recycle through an external heat exchanger. Temperature and level in the reactor, as well as the recycle flow rate, are controlled by feedback control systems (cascade control for the case of temperature).

4.5.1 Modelling of the CSTR system

A dynamic model of the CSTR system is developed using some results presented in Franks (1972). The model is used to simulate the process and serves as a test bed for several fault diagnosis systems. Several assumptions have been made in modelling the system and, hence, the developed model may not be very accurate. It is assumed that perfect mixing takes place in the reactor and perfect heat exchange takes place in the heat exchanger. To simplify the model, it is also assumed that the reactant and the product have the same density and specific heat. The model is developed based on mass and heat balances in the process and is listed below:

Table 4.3: Performance under multiple failures

faults initiated	fault diagnosed
hot water control valve fail cold water control valve fail	cold water control valve fail
level sensor 1 fail temp. sensor 1 fail	temp. sensor 1 fail
level sensor 2 fail temp. sensor 2 fail	temp. sensor 2 fail
hand valve 2 blocked cold water control valve fail	hand valve 2 blocked
temp. sensor 1 fail level sensor 2 fail hot water control valve fail	temp. sensor 1 fail

$$A\frac{dH}{dt} = Q_1 + Q_2 - Q_3 \tag{4.11}$$

$$AH\frac{dC_a}{dt} = Q_1(C_{a0} - C_a) - r_aAH \tag{4.12}$$

$$AH\frac{dC_b}{dt} = r_aAH - C_bQ_1 \tag{4.13}$$

$$AHB_2\frac{dT}{dt} = B_1Q_1(T_1 - T) - B_2Q_2(T - T_2) + H_rr_a \tag{4.14}$$

$$B_1 = C_{a0}\rho C + (1 - C_{a0})\rho_0C_0 \tag{4.15}$$

$$B_2 = \rho C(C_a + C_b) + (1 - C_a - C_b)\rho_0C_0 \tag{4.16}$$

$$r_a = K_rC_a^n \quad (n > 0) \tag{4.17}$$

$$K_r = a_re^{-b_r/T} \tag{4.18}$$

$$Q_2 = K_2 A_2 \sqrt{P} \tag{4.19}$$

$$Q_4 = K_4 A_4 \sqrt{P} \tag{4.20}$$

$$Q_3 = Q_2 + Q_4 \tag{4.21}$$

$$P = P_0 + \Delta P \tag{4.22}$$

$$P_0 = H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \tag{4.23}$$

$$Q_5 = K_5 A_5 \sqrt{P_5} \tag{4.24}$$

$$T_2 = \frac{C_0 \rho_0 Q_5 T_5 + Q_2 T[C\rho(C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]}{C_0 \rho_0 Q_5 + Q_2 [C\rho(C_a + C_b) + C_0 \rho_0 (1 - C_a - C_b)]} \tag{4.25}$$

where

H = level in the reactor (cm)

T = temperature in the reactor ($^\circ C$)

A = cross-sectional area of the reactor (cm^2)

Q_1 = flow rate of input reactant (cm^3/Sec)

Q_2 = flow rate of the recycled reactant (cm^3/Sec)

Q_3 = flow rate of the liquid leaving the reactor (cm^3/Sec)

C_a = concentration of reactant in the reactor

C_b = concentration of product in the reactor

C_{a0} = concentration of reactant in the input stream

r_a = reaction rate (g/Sec)

H_r = reaction heat constant (KJ/g)

T_1 = temperature of input reactant ($^\circ C$)

T_2 = temperature of the recycled reactant after heat exchange ($^\circ C$)

ρ = density of the reactant (g/cm^3)

C= specific heat of the reactant ($J/g^\circ C$)

ρ_0 = density of the solvent (g/cm^3)

C_0 = specific heat of the solvent ($J/g^\circ C$)

K_r = reaction rate constant (g/Sec)
a_r = constant peculiar to reaction (g/Sec)
b_r = constant peculiar to reaction $(°C)$
K_2 = restriction parameter of valve 3 $(cm^4/g^{1/2}Sec)$
A_2 = fractional opening of valve 3
P = pressure of liquid leaving the pump (g/cm^2)
Q_4 = flow rate of the product (cm^3/Sec)
K_4 = restriction parameter of valve 1 $(cm^4/g^{1/2}Sec)$
A_4 = fractional opening of valve 1
P_0 = pressure at the bottom of the reactor (g/cm^2)
ΔP = pressure increase caused by pump (g/cm^2)
T_5 = temperature of cold water entering heat exchanger $(°C)$
Q_5 = flow rate of cold water entering heat exchanger (cm^3/Sec)
K_5 = restriction parameter of valve 2 $(cm^4/g^{1/2}Sec)$
A_5 = fractional opening of valve 2
P_5 = pressure of feed cold water to the heat exchanger (g/cm^2)

The model parameters and the nominal values of certain process variables are given in Table 4.4. The controllers used are PI controllers of the form

$$u(t) = K(e(t) + \frac{\sum_{i=1}^{t} e(i)}{T_i})$$

where $u(t), e(t), K$, and T_i are the control signal, error signal, controller gain, and integration time respectively. The parameters of the controllers, together with the setpoints of the controlled variables, are given in Table 4.5.

4.5.2 Formulation of diagnostic rules

The CSTR system is decomposed into three subsystems. The first subsystem, S_1, is the external reactant feed subsystem, which includes pipe 1 and associated sensors. The second subsystem, S_2, is the reaction subsystem including the reaction vessel, pipe 2, pump, pipe 3, valve 1, pipe 11 and associated sensors. The remaining components form the third subsystem, S_3, which is the heat exchange subsystem. The diagnosis graph corresponding to this decomposition is shown in Figure 4.8.

The Connection Matrix is

Table 4.4: Model parameters and nominal values of certain process variables

Parameters or variables	values
A	$300.0cm^2$
a_r	$0.8g/Sec$
b_r	$66.9°C$
H_r	$430KJ/g$
K_2	$3.26cm^4/g^{1/2}Sec$
K_4	$4.34cm^4/g^{1/2}Sec$
K_5	$4.7cm^4/g^{1/2}Sec$
Q_1	$300.0cm^3/Sec$
T_1	$20°C$
C_{a0}	0.8
ρ	$1.2g/cm^3$
ρ_0	$1.1g/cm^3$
C	$0.9J/g°C$
C_0	$0.8J/g°C$
P_5	$200.0g/cm^2$
T_5	$20°C$
ΔP	$200.0g/cm^2$

Table 4.5: Controller parameters and set points

Control loops	Setpoints	Controller parameters	
		K	T_i
H	$30.0cm$	6.0	10.0
Q_2	$200.0cm^3/Sec$	0.2	3.0
T	$50.0°C$	4.0^{*}	8.0^{*}
		0.1^{**}	6.0^{**}

* Primary control loop

** Secondary control loop

$$C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{4.26}$$

The Self-Causal Matrix for the first subsystem is

$$CS_1 = \begin{array}{c} \\ Q_1 \\ T_1 \\ C_{a0} \end{array} \begin{array}{c} \begin{array}{ccc} Q_1 & T_1 & C_{a0} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{4.27}$$

which suggests that the three measurements in the first subsystem cannot affect each other. The labels on the top and the left of the matrix, Q_1, T_1, and C_{a0}, are the flow rate, temperature, and concentration of the external feed reactant respectively.

The Self-Causal Matrix for the second subsystem is

$$CS_2 = \begin{array}{c} \\ H \\ T \\ Q_4 \\ C_a \\ CQ_4 \end{array} \begin{array}{c} \begin{array}{ccccc} H & T & Q_4 & C_a & CQ_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \tag{4.28}$$

The labels on the top and the left of the matrix, H, T, Q_4, C_a, and CQ_4, are level and temperature in the reactor, flow rate through valve 1, concentration of the reactant in the product, and controller output to valve 1 respectively.

The Self-Causal Matrix for the third subsystem is

$$CS_3 = \begin{array}{c} \\ Q_2 \\ CQ_2 \\ Q_5 \\ CQ_5 \\ CT \\ P \\ T_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} Q_2 & CQ_2 & Q_5 & CQ_5 & CT & P & T_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{4.29}$$

The labels on the top and the left of the matrix, $Q_2, CQ_2, Q_5, CQ_5, CT, P$, and T_5, are the flow rate through valve 3, controller output to valve 3, flow rate through valve 2, controller output to valve 2, prime controller output of the cascade controller, pressure and temperature of the cold water to the heat exchanger respectively.

The Measurement Causal Matrix from S_1 to S_2 is

$$CM_{12} = \begin{array}{c} \\ Q_1 \\ T_1 \\ C_{a0} \end{array} \begin{array}{c} \begin{array}{ccccc} H & T & Q_4 & C_a & CQ_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{4.30}$$

The Measurement Causal Matrix from S_2 to S_3 is

$$CM_{23} = \begin{array}{c|ccccccc} & Q_2 & CQ_2 & Q_5 & CQ_5 & CT & P & T_5 \\ \hline H & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ Q_4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ CQ_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{4.31}$$

The Measurement Causal Matrix from S_3 to S_2 is

$$CM_{32} = \begin{array}{c|ccccc} & H & T & Q_4 & C_a & CQ_4 \\ \hline Q_2 & 1 & 0 & 1 & 0 & 0 \\ CQ_2 & 0 & 1 & 0 & 0 & 0 \\ Q_5 & 0 & 0 & 0 & 0 & 0 \\ CQ_5 & 0 & 0 & 0 & 0 & 0 \\ CT & 0 & 0 & 0 & 0 & 0 \\ P & 0 & 0 & 0 & 0 & 0 \\ T_5 & 0 & 1 & 0 & 0 & 0 \end{array} \tag{4.32}$$

The Fault Measurements Matrix for S_1 is

$$FM_1 = \begin{array}{c} \\ P1 \\ FT \\ FQ1 \\ FCa0 \\ SQ1 \\ ST1 \\ SCa0 \end{array} \begin{array}{c} \begin{array}{ccc} Q_1 & T_1 & C_{a0} \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \end{array} . \tag{4.33}$$

The labels on the left of the matrix, $P1, FT, FQ, FCa0, SQ1, ST1$, and $SCa0$, stand for pipe 1 blockage, feed reactant temperature, flow rate, and concentration abnormal, sensor failures of Q_1, T_1, and C_{a0} respectively.

The Fault Measurement Matrix for S_2 is

$$FM_2 = \begin{array}{c} \\ P2 \\ P10 \\ V1 \\ SH \\ ST \\ SQ4 \\ SCa \\ CO4 \end{array} \begin{array}{c} \begin{array}{ccccc} H & T & Q_4 & C_a & CQ_4 \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{4.34}$$

The labels on the left of the matrix, $P2, P10, V1, SH, ST, SQ4, SCa$, and $CO4$, stand for blockages of pipes 2 and 3, valve 1 fails high, sensor failures for H, T, Q_4, and C_a, and level controller failure respectively.

The Fault Measurements Matrix for S_3 is

$$FM_3 = \begin{array}{c|ccccccc|} & Q_2 & CQ_2 & Q_5 & CQ_5 & CT & P & T_5 \\ P4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ P7 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ V2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ V3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ Tc & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ P & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ SQ2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ SQ5 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ STc & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ SP & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ CO2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \qquad (4.35)$$

The labels on the left of the matrix, $P4, P7, V2, V3, Tc, P, SQ2, SQ5, STc, SP$, and $CO2$, stand for pipe 4 blockage, pipe 7 blockage, valve 2 fails high, valve 3 fails high, feed cold water temperature and pressure abnormal, sensor failures for Q_2, Q_5, Tc, and P, and Q_2 controller failure respectively.

Abnormal behaviour detection is similar to that for the mixing process. A set of error tolerances are defined for controlled variables and if any controlled variable exceeds its tolerance then abnormal behaviour is indicated. A set of varying ranges are defined for other variables and abnormal behaviour is indicated if any variable exceeds its varying range. Manipulated variables should change in the same direction as the corresponding controller outputs and if they do not, abnormal behaviour is identified.

Diagnostic rules are formulated from the knowledge on system structures and component functions in a similar way as that for the mixing process. The diagnosis system is defined by a main problem CSTRD and 11 subproblems. The rule file for the main problem is shown in Figure 4.9. The function of the main problem is to classify the observed abnormalities. The outcomes of the main problem are several subproblems each corresponding to a type of abnormality.

The subproblem RLEVEL will be used if abnormalities are present in the measurement of level in the reactor and its rules are shown in Figure 4.10. The definitions of attributes used in CSTRD and RLEVEL are given in Table 4.6. These rules are developed from the following considerations. There are two situations when the level in the reactor is abnormal, one is that the level is higher than its set point and another is that the level is lower than its set point. Consider the first situation.

From Eq(4.28) it can be seen that only Q_4 in S_2 can affect H. If Q_4 is high, then it is responsible for abnormality in H and is activated. If Q_4 is activated, then Eq(4.28) suggests that only CQ_4 in S_2 can affect Q_4. If CQ_4 is high then it is responsible for Q_4 being high and is activated. If CQ_4 is activated then Eq(4.28) shows that H can affect CQ_4 and, since H is lower than its set point, it is not responsible for CQ_4 being high. Therefore, Eq(4.30) and Eq(4.32) show that no variable can affect CQ_4. In this case, Eq(4.34) suggests that the only failure would be controller failure. If Q_4 is activated and CQ_4 is not then, from Eq(4.30) and Eq(4.32), it can be seen that no other variables in S_1 or S_3 can affect Q_4, and Eq(4.34) suggests that pipe 2 blockage, pipe 10 blockage control valve 1 fails high, and sensor Q_4 failure are the candidate failures. The first two failures can be ruled out as Q_4 is high and pipe 2 or pipe 10 blockage cannot cause Q_4 being high. The last one can be ruled out by the single failure assumption as it will not cause abnormality in H and, therefore, the failure is control valve 1 fails high. If Q_4 is not activated, then Eq(4.30) suggests that Q_1 in S_1 can affect H. If Q_1 is low then it is activated and in this case, Eq(4.33) shows that the candidate failures are pipe 1 blockage, external feed reactant flow rate abnormal, and sensor Q_1 failure. Sensor failure is again ruled out by the single failure assumption and the diagnosis result at this stage would be pipe 1 blockage or feed reactant flow low. If Q_1 is not activated, then Eq(4.32) shows that only Q_2 in S_3 can affect H, and if Q_2 is low then it is responsible for H being low and is activated. If Q_2 is activated then Eq(4.29) shows that CQ_2 in S_2 can affect Q_2, and CQ_2 will be activated if it is low. If CQ_2 is activated, then Eq(4.29) and Eq(4.31) suggest that only Q_2 can affect CQ_2, but Q_2 will not be responsible since Q_2 is low. In this case, Eq(4.35) suggests that the only failure would be controller failure. If only Q_2 is activated then Eq(4.35) suggests that the failures would be pipe 2 blockage or sensor Q_2 failure and the last is ruled out by the single failure assumption. If only H is activated, then Eq(4.34) suggests that the candidate failures would be pipe 2 blockage, pipe 10 blockage, control valve 1 fails high, and sensor failure. The first two failures can be ruled out as they will not cause H to increase and the third one can also be ruled out as it will cause Q_4 to be high, which is not observed. Therefore, the only possible failure is sensor H failure. The formulation of rules for the situation where H is higher than its set point is similar to the above.

The developed diagnosis system has been tested on the simulation of the CSTR system. In the simulation, the sampling time is 4 seconds and the diagnosis system collects and examines process data every 20 seconds during normal operation and every 4 seconds when abnormal behaviour is detected. The possible faults that may occur are initiated separately during simulation and they were diagnosed quite

Table 4.6: Definitions of attributes used in CSTRD and RLEVEL

Attributes	Definitions
ABS2	Subsystem 2 is abnormal
ABS3	Subsystem 3 is abnormal
ABH	Reactor level is abnormal
ABT	Reactor temperature is abnormal
ABQ1	Q_1 is abnormal
ABQ2	Q_2 is abnormal
ABQ4	Q_4 is abnormal
ABQ5	Q_5 is abnormal
ABT1	T_1 is abnormal
ABT5	T_5 is abnormal
HLTSP	Reactor level is lower than its setpoint
Q1LO	Feed reactant flow low
Q1HI	Feed reactant flow high
Q2LO	Q_2 is low
CQ2LO	Controller output to valve 3 is low
Q4LO	Q_4 is low
Q4HI	Q_4 is high
CQ4LO	Controller output to valve 1 is low
CQ4HI	Controller output to valve 1 is high

successfully.

4.6 Conclusions

A method for formulating diagnostic rules from knowledge of system structures and component functions has been developed. Based on this deep knowledge, diagnosis can be performed hierarchically, and it is shown that structural decomposition can rapidly narrow diagnostic focus. Since structural decomposition corresponds to plant topology, it could be easier to implement. Advantages of a rule based format are that rules are efficient to evaluate and diagnostic rules can be combined with other rules pertaining to plant operations. The successful application of this method in developing diagnosis systems for the pilot scale mixing process and a simulated CSTR system suggests that this method provides a systematic and efficient means for the design of on-line rule based fault diagnosis systems.

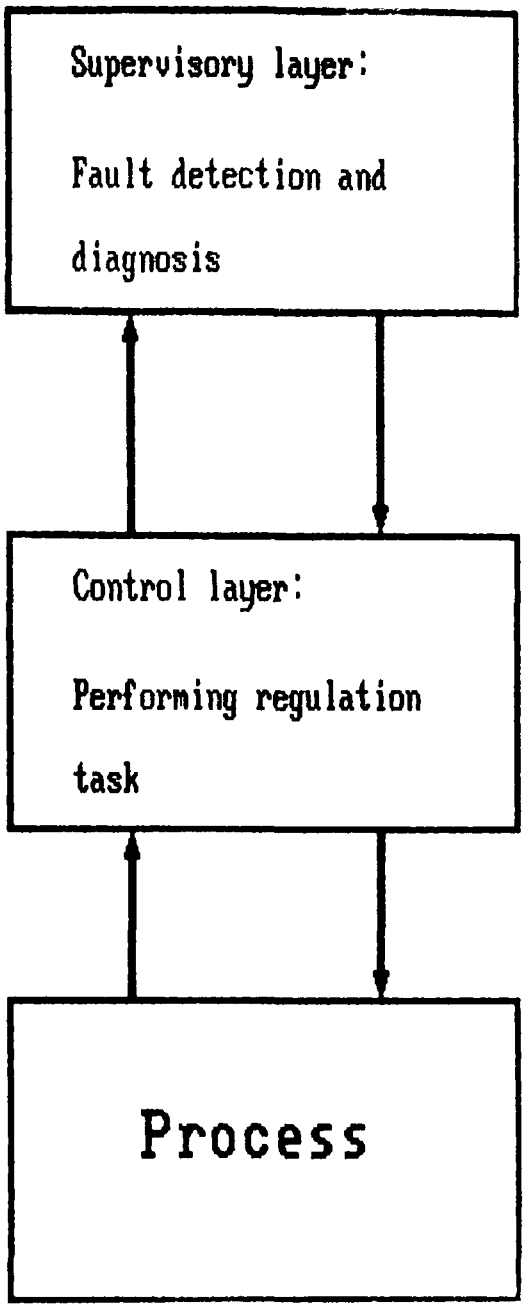


Figure 4.1 A hierarchical control system

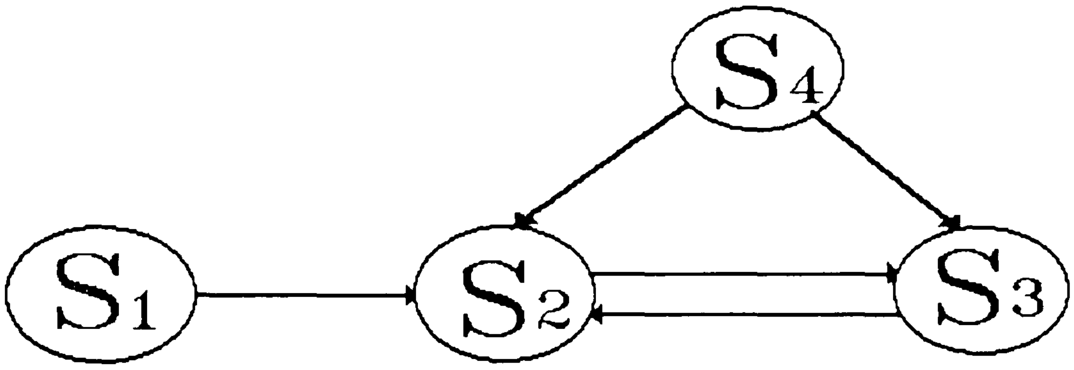


Fig.4.2 A diagnosis graph

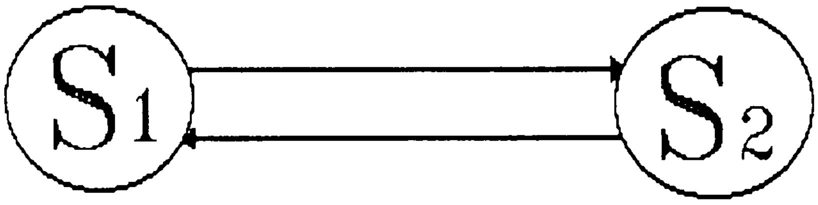


Figure 4.3 Diagnosis graph for the mixing process

```
@tdiag
[abs2    ] :
     n : [abt1   ] :
            n : examining if sensor h1 failed (sensh1)
            y : examining if sensor t1 failed (senst1)
     y : [abt2   ] :
            n : performing subproblem level2 (level2)
            y : performing subproblem temp2 (temp2)
```

Figure 4.4 (a). Diagnostic rules for main problem TDIAG

```
@temp2
[t2ltsp  ] :
     y : [t1dcr   ] :
             y : [qhdcr   ] :
                   y : controller failure
                   n : hot water control valve failure
                       giving low output
             n : examining if sensor t2 failed (senst2)
     n : [t1inc   ] :
             n : examining if sensor t2 failed (senst2)
             y : [qhinc   ] :
                   n : hot water control valve failure
                       giving high output
                   y : controller failure
```

Figure 4.4 (b). Diagnostic rules for subproblem TEMP2

```
@level2
[h2ltsp  ] :
     y : [h1dcr   ] :
         y : [qcdcr   ] :
               y : controller failure
               n : cold water control valve failure
                   giving low output
         n : [h2cond  ] :
               y : hand valve 1 is blocked
               n : examining if sensor h2 failed (sensh2)
     n : [h1inc   ] :
         n : [h2coni  ] :
               y : hand valve 2 is blocked
               n : examining if sensor h2 failed (sensh2)
         y : [qcinc   ] :
              y : cold water control valve failure
                  giving high output
              n : controller failure
```

Figure 4.4 (c). Diagnostic rules for subproblem LEVEL2

```
@senst2
[t2sc    ] :
     y : sensor t2 failure
     n : [t2ltsp  ] :
             y : [t1it2s  ] :
                          y : sensor t2 failure
                          n : no failure found so far
             n : [t1dt2s  ] :
                          y : sensor t2 failure
                          n : no failure found so far
```

Figure 4.4 (d). Diagnostic rules for subproblem SENST2

```
@sensh2
[h2sc    ] :
     y : sensor h2 failure
     n : [h2ltsp  ] :
             y : [h1ih2s  ] :
                          y : sensor h2 failure
                          n : no failure found so far
             n : [h1dh2s  ] :
                          y : sensor h2 failure
                          n : no failure found so far
```

Figure 4.4 (e). Diagnostic rules for subproblem SENSH2

```
@senst1
[t1sc    ] :
     y : sensor t1 failure
     n : [t2nrsp  ] :
             y : [dth     ] :
                          y : sensor t1 failure
                          n : no failure found so far
             n : no failure found so far
```

Figure 4.4 (f). Diagnostic rules for subproblem SENST1

```
@sensh1
[h1sc    ] :
     y : sensor h1 failure
     n : [h2nrsp  ] :
             y : [dhh     ] :
                         y : sensor h1 failure
                         n : no failure found so far
             n : no failure found so far
```

Figure 4.4 (g). Diagnostic rules for subproblem SENSH1

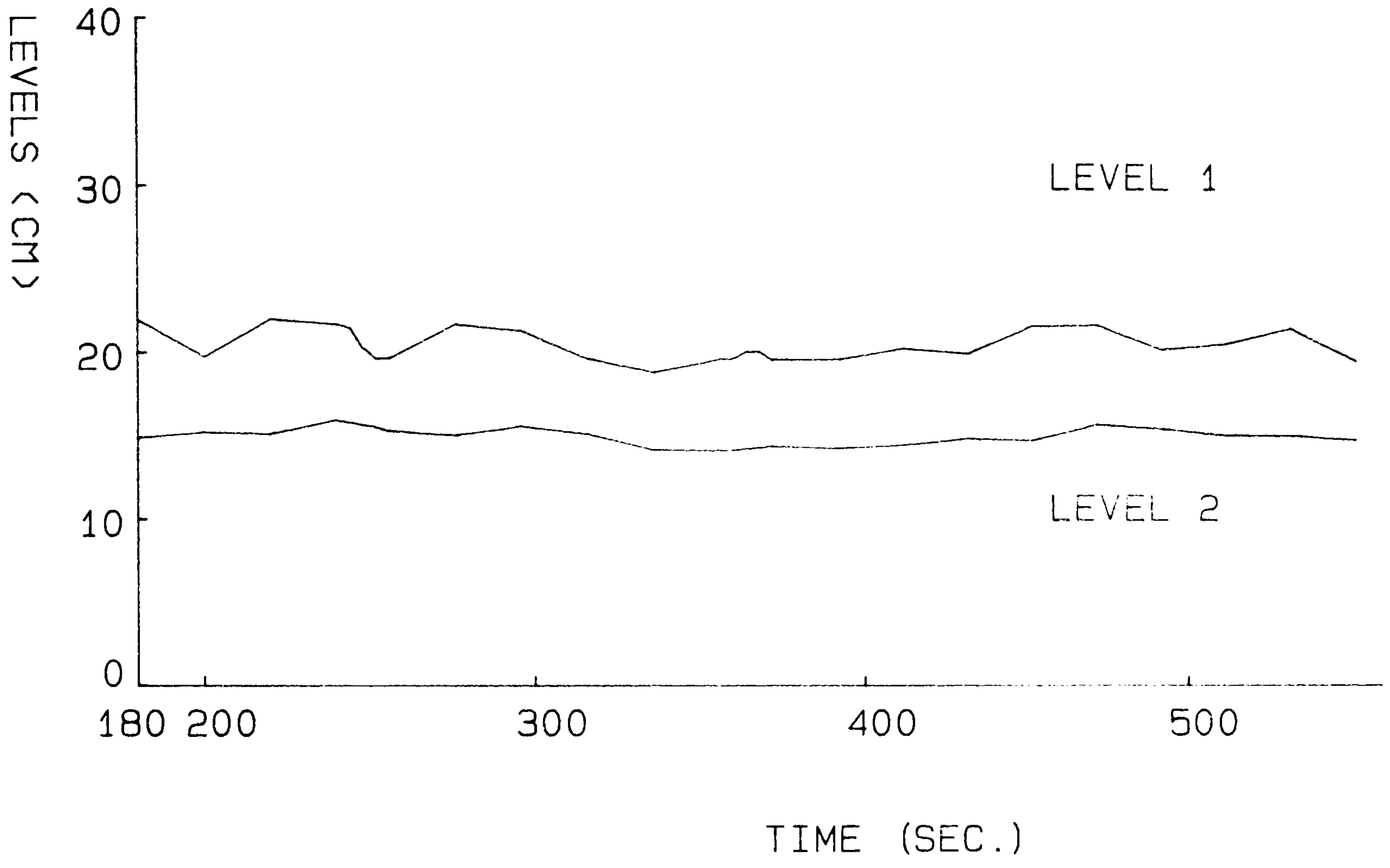


Figure 4.5(a) On-line level measurements

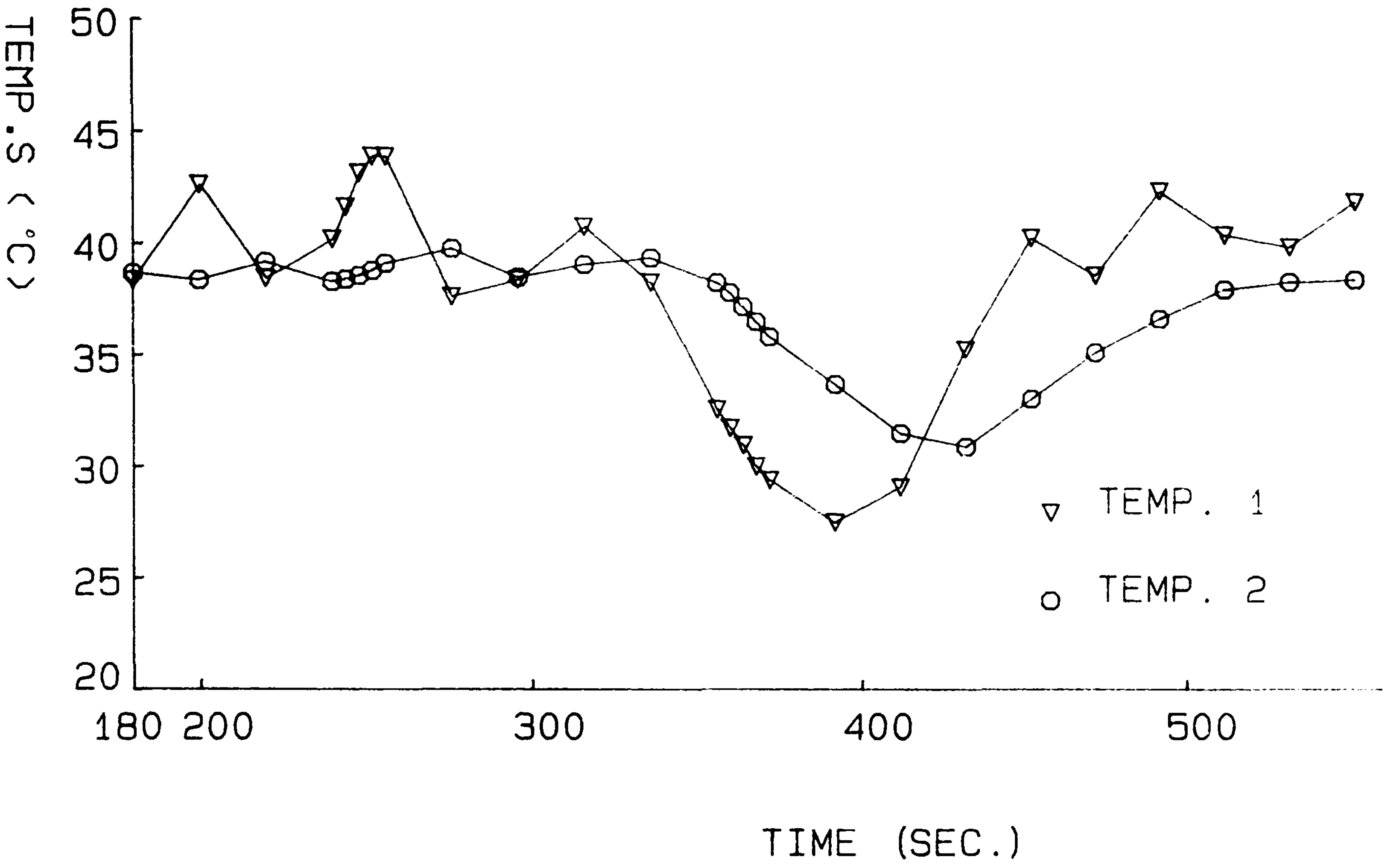


Figure 4.5(b) On-line temperature measurements

```
THE CURRENT PROBLEM SUITE IS { tanksd }
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


A DECISION WAS REACHED FOR PROBLEM {tdiag }
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Since
subsystem 2 is abnormal
    the decision cannot be any of :
      examining if sensor h1 failed
      examining if sensor t1 failed
and temperature 2 is abnormal
    the decision cannot be :
      performing subproblem level2

Hence, the decision is

 performing subproblem temp2


A DECISION WAS REACHED FOR PROBLEM {temp2 }
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Since
t2 is lower than its setpoint
    the decision cannot be :
      hot water control valve failure giving high output
and t1 is decreasing
    the decision cannot be :
      examining if sensor t2 failed
and qh is not decreasing
    the decision cannot be :
      controller failure

Hence, the decision is

 hot water control valve failure giving low output
```

Figure 4.6 Reasoning procedures of the diagnosis system

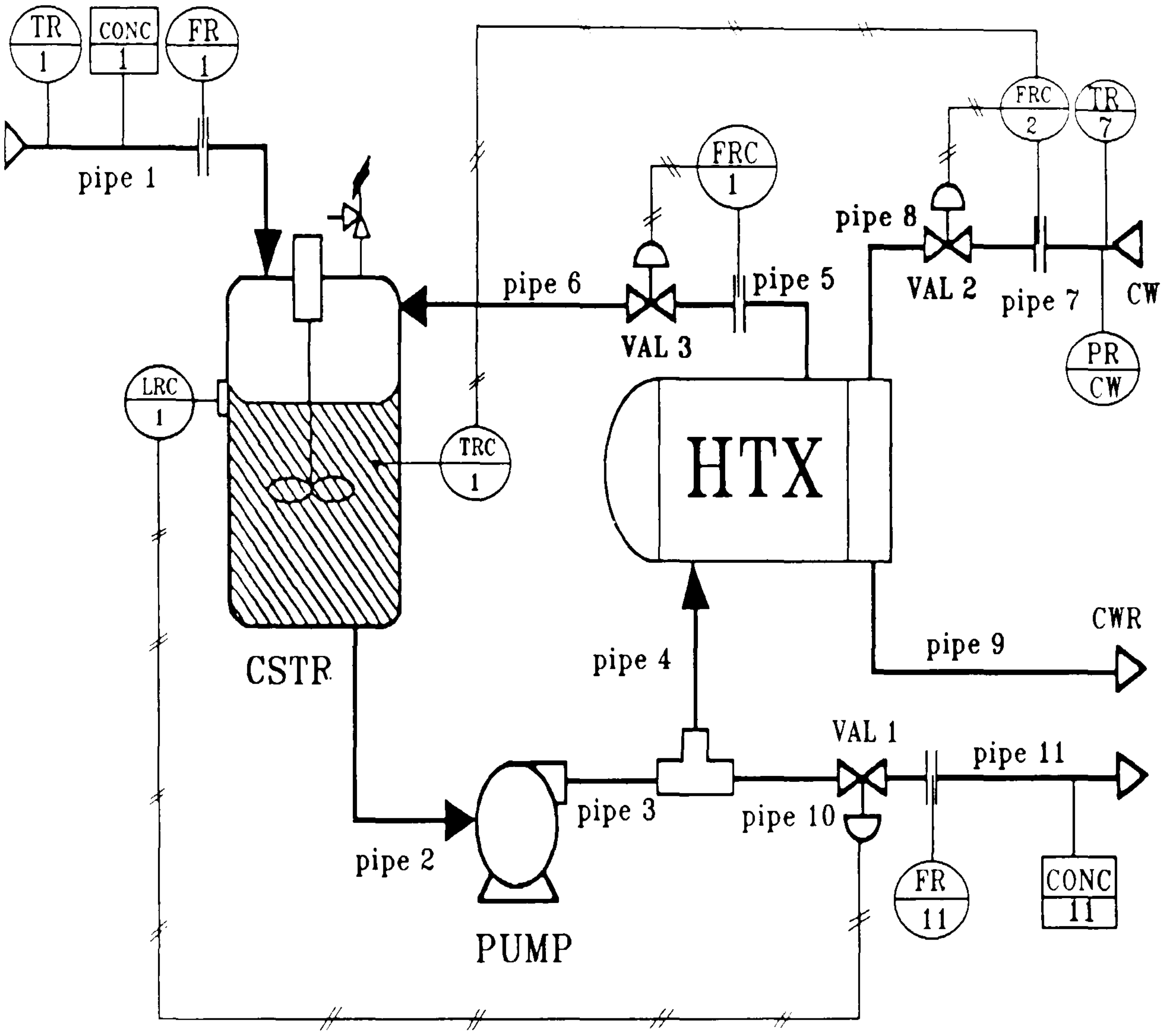


Fig.4.7 Continuous stirred tank reactor with recycle

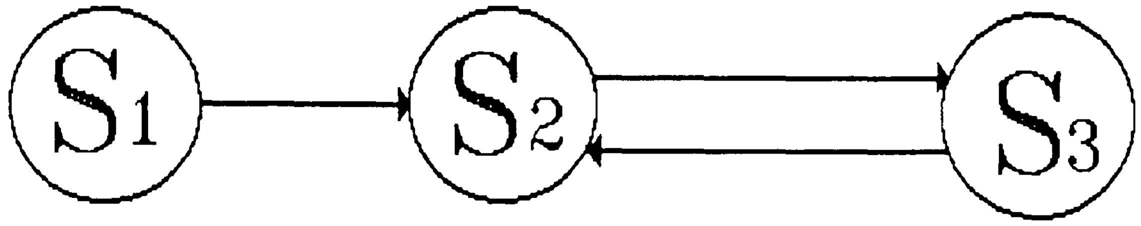


Figure 4.8 Diagnosis graph for the CSTR system

```
@cstrd.rul
[abs2    ] :
     y : [abh     ] :
           y : perform subproblem rlevel
           n : [abt     ] :
                y : perform subproblem rtemp
                n : [abq4    ] :
                     y : perform subproblem flow4
                     n : perform subproblem ca
     n : [abs3    ] :
           y : [abq2    ] :
                y : perform subproblem flow2
                n : [abq5    ] :
                     y : perform subproblem flow5
                     n : [abt5    ] :
                             y : perform subproblem temp5
                             n : perform subproblem pres
           n : [abq1    ] :
                 y : perform subproblem flow1
                 n : [abt1    ] :
                         y : perform subproblem temp1
                         n : perform subproblem ca0
```

Figure 4.9 Diagnostic rules for the main problem CSTRD

```
@rlevel.rul
[hltsp   ] :
     y : [q4hi    ] :
           y : [cq4hi   ] :
                 y : controller Q4 failure
                 n : control valve 1 fails high
           n : [q1lo    ] :
                 y : pipe 1 is blocked
                 n : [q2lo    ] :
                        y : [cq2lo   ] :
                              y : controller Q2 failure
                              n : pipe 2 is blocked
                        n : sensor H failure
     n : [q4lo    ] :
           y : [cq4lo   ] :
                 y : controller Q4 failure
                 n : [q2lo    ] :
                        y : pipe 2 is blocked
                        n : pipe 10 is blocked
           n : [q1hi    ] :
                 y : external feed reactant flow high
                 n : sensor H failure
```

Figure 4.10 Diagnostic rules for subproblem RLEVEL

Chapter 5

On-line fault diagnosis based on qualitative simulation

5.1 Introduction

The previous chapter describes a deep knowledge based approach, causal search, which is based on knowledge of system structure and component functions. One of the frequently used human diagnostic strategies is the hypothesis-test strategy (Rasmussen 1980). From the observed patterns of abnormalities, the operator hypotheses a potential cause of the upset and then mentally simulates the effect of the hypothesized malfunction on process behaviour. If the simulated behaviour matches the observed one, the hypothesis is retained, otherwise, an alternative hypothesis may be formed. The procedure can be implemented automatically on a computer using the recently developed qualitative reasoning techniques (Bobrow 1984). When the monitored process contains a large number of variables, the qualitative reasoning method may be used to simplify the computation. The qualitative reasoning method is also appropriate as it depends less on accurate quantitative information. This is particularly useful in simulating the effect of a fault as the exact severity of a fault is generally not known.

There are several different approaches in qualitative reasoning such as de Kleer and Brown's confluence based qualitative reasoning (de Kleer and Brown 1984), Forbus' qualitative process theory (Forbus 1984), and Kuipers' qualitative simulation (Kuipers 1986). In this research, de Kleer and Brown's confluence based qualitative reasoning method is used. The qualitative model used in this approach is a set of confluences which are qualitative equations and are derived from a quantitative model of the process under concern. This would be suitable for process control as a quantitative model of a process can usually be developed. A further advantage of this approach is that the effect of a fault can be easily represented by the deviation of the corresponding process variable. For example, a blockage or a partial blockage of a valve can be represented by a decrease in the opening area of the valve in the form

$$\delta A = -$$

where A is the opening area of the valve. By setting δA in the model to "$-$", the model can be used to simulate the process under this fault. Therefore, the qualitative model can easily be used to simulate the process under normal or various faulty conditions.

However, due to the lack of quantitative information, ambiguity often occurs in qualitative reasoning, especially when a large number of qualitative variables are involved. This ambiguity can be reduced by taking account of the order of magnitude of different variables. Raiman (1986) has proposed a method of order of magnitude reasoning to reduce the ambiguity, but his method only reduces the ambiguity in some specific cases where some variables' magnitudes are negligible compared with those of others. In this research, a method for reducing ambiguity in more general cases by taking account of the relative magnitude relations among variables has been investigated.

The diagnostic strategy used in this chapter is the "hypothesis-test strategy" (Rasmussen 1980, Moor and Kramer 1986). Unlike the failures of other components, the effect of sensor failures on process behaviour cannot be easily represented and, therefore, sensor failures are treated differently from other failures. Since the diagnosis systems described in this chapter are real-time diagnosis systems based on on-line measurements, it would be necessary to determine whether sensors are working normally before considering other components. Thus, when generating a hypothesis, sensor failures are considered first. If a hypothesis is a sensor failure, then it is confirmed or denied by a set of heuristics relating to the diagnosis of sensor failures. If a hypothesis is the failure of other components rather than sensors, then the diagnosis system will predict the behaviour of the process under this hypothesis and compare the prediction with the actual measurements. The hypothesis is confirmed if the actual behaviour follows the predicted behaviour, otherwise, it is denied.

In the next section, a brief review of confluence based qualitative reasoning is given and is followed by a new approach for reducing ambiguity in qualitative reasoning. Section 5.3 describes the use of the qualitative reasoning approach to solve a two mass collision problem, which suggests that the proposed qualitative reasoning approach can produce a better solution than that of Raiman (1986). Qualitative modelling of the mixing process and the development of a fault diagnosis system based on qualitative reasoning for the mixing process is presented in Section 5.4. Section 5.5 describes the development of a fault diagnosis system based on qualitative simulation for the CSTR system. The last section contains some concluding remarks.

5.2 Qualitative reasoning

5.2.1 Qualitative reasoning based on confluence

De Kleer and Brown (de Kleer and Brown 1984) discuss a qualitative reasoning method based on confluences. This method is also referred to as Incremental Qualitative Analysis (IQA) (Herbert and Williams 1986, 1987). Since one of the most important features of a physical variable is whether it is increasing, decreasing, or unchanging; $+$, $-$ and 0 are defined as the quantity space where $+$, $-$ and 0 represent the cases that a variable is increasing, decreasing, and unchanging respectively. More generally, the qualitative value of a physical variable X corresponding to a specified landmark value a is denoted as $[X]_a$ and

$$[X]_a = \begin{cases} +, & \text{if } X > a, \\ 0, & \text{if } X = a, \\ -, & \text{if } X < a. \end{cases}$$

Usually the landmark value used is 0, and $[X]_0$ is denoted as $[X]$ for simplicity. For practical applications, such as fault diagnosis, threshold values are defined for the conversion from quantitative values to qualitative values, such that

$$[X] = \begin{cases} +, & \text{if } X > X_+, \\ 0, & \text{if } X_- \leq X \leq X_+, \\ -, & \text{if } X < X_-. \end{cases}$$

where X_+ and X_- are the threshold values for the physical variable X.

Addition and multiplication of qualitative variables are defined in Table 5.1 and Table 5.2 respectively. In Table 5.1, "?" stands for unknown, it may be any one of the values: $+, 0$, and $-$.

Table 5.1: Addition of qualitative variables $[A]$ and $[B]$

$[A]$	$[B]$		
	$-$	0	$+$
$-$	$-$	$-$	$?$
0	$-$	0	$+$
$+$	$?$	$+$	$+$

Table 5.2: Multiplication of qualitative variables $[A]$ and $[B]$

$[A]$	$[B]$		
	$-$	0	$+$
$-$	$+$	0	$-$
0	0	0	0
$+$	$-$	0	$+$

The qualitative behaviour of a physical system can be described by a set of confluences which are formally derived from the quantitative equations for the system. This ensures that the qualitative model is consistent with the quantitative one.

From Table 5.1 it can be seen that the addition of two qualitative variables with opposite values $+$ and $-$ is unknown. Ambiguity is a major problem associated with qualitative reasoning. Ambiguity is due to the lack of quantitative information and, with the addition of some available quantitative information, this ambiguity may be reduced. Raiman (1986) investigates using order of magnitude reasoning to reduce ambiguity. As suggested by Oyeleye and Kramer (1988), ambiguity could also be reduced by qualitative constraints derived from redundant numerical equations. For example, considering the following two equations

$$X_1 + X_2 - X_3 = 0$$

$$2X_1 + X_2 + X_3 = 0$$

from which the following qualitative constraints can be derived.

$$[X_3] = [X_1] + [X_2]$$

$$[X_1] = [X_3] - [X_2]$$

$$[X_2] = [X_3] - [X_1]$$

Table 5.3: Solution of $[X_2]$ and $[X_3]$

possible solutions			viable
$[X_1]$	$[X_2]$	$[X_3]$	
+	+	+	No
+	+	0	No
+	+	−	No
+	0	+	No
+	0	0	No
+	0	−	No
+	−	+	Yes
+	−	0	Yes
+	−	−	Yes

$$[X_3] = -[X_1] - [X_2]$$
$$[X_1] = -[X_2] - [X_3]$$
$$[X_2] = -[X_1] - [X_3]$$

Suppose that $[X_1] = +$, then the solutions for $[X_2]$ and $[X_3]$ from the above qualitative constraints are provided in Table 5.3, from which it can be seen that the solution for $[X_3]$ is ambiguous. A redundant numerical equation can be derived by substracting the first numerical equation from the second and is given bellow:

$$X_1 + 2X_3 = 0$$

from which an additional qualitative constraint can be obtained as

$$[X_1] = -[X_3]$$

which gives an unambiguous solution $[X_3] = -$.

5.2.2 Order of magnitude reasoning

To reduce the ambiguity in qualitative reasoning, Raiman (1986) developed a formal system FOG which takes account of the information on the order of magnitude of physical variables to remove ambiguity. In FOG, three operators, Ne, Vo, and Co, are defined to represent the order of magnitude relations between physical variables such that

$A\ Ne\ B$ stands for A is negligible in relation to B,

$A\ Vo\ B$ stands for A is close to B,

$A\ Co\ B$ stands for A has the same sign and order of magnitude as B.

To perform qualitative reasoning, 31 inference rules are defined. From the three defined operators, it can be seen that this method can only reduce ambiguity in some specific cases where some variables' magnitudes are negligible to those of other variables. This can be illustrated by a simple example. Consider the situation where $[A] = -[B]$ and $A\ Ne\ B$. The addition of $[A]$ and $[B]$ will be $[B]$ instead of unknown. However, if the relation between $[A]$ and $[B]$ is not "negligible", then ambiguity cannot be removed.

Here a new approach which can reduce ambiguity in more general cases is introduced. In this approach, four operators, Rmh, Rmc, Rml, and Rmn, are defined such that

$A\ Rmh\ B$ stands for the relative order of magnitude of A is higher than that of B,

$A\ Rmc\ B$ stands for the relative order of magnitude of A is close to that of B,

$A\ Rml\ B$ stands for the relative order of magnitude of A is lower than that of B,

$A\ Rmn\ B$ stands for the relative order of magnitude of A is negligible to that of B.

Note in the above definitions, Rmn is a subclass of Rml and the relations between two variables can only be either Rmh, Rmc, or Rml.

To perform qualitative reasoning, the following 18 inference rules are defined:

$$R1: \quad A\ Rmh\ B \Leftrightarrow B\ Rml\ A$$

$$R2: \quad A\ Rmc\ B \Rightarrow B\ Rmc\ A$$

$$R3: \quad A\ Rmn\ B \Rightarrow A\ Rml\ B$$

$$R4: \quad A * B,\ B * C \Rightarrow A * C$$

(* stands for any operators)

$$R5:\ A\ Rmc\ B,\ B*C \Rightarrow A*C$$

$$R6:\ A\ Rmh\ B,\ C\ Rmn\ B \Rightarrow C\ Rmn\ A$$

$$R7:\ A\ Rmh\ B \Rightarrow [A]+[B]=[A]$$

$$R8:\ A\ Rmc\ B,\ [A]=-[B] \Rightarrow [A]+[B]=0$$

$$R9:\ [A]=[B]+[C],\ [A]=-[B] \Rightarrow [C]=[A],\ C\ Rmh\ B$$

$$R10:\ A\ Rmc\ B,\ C*D \Rightarrow A\cdot C*B\cdot D$$

$$R11:\ A*B,\ C*D \Rightarrow A\cdot C*B\cdot D$$

$$\begin{aligned}R12:\ & (A+B)\ Rmc\ (C+D),\ [A+B]=[C+D],\\ & [A]=[C],\ A\ Rmc\ C \Rightarrow [B]=[D],\ B\ Rmc\ D\end{aligned}$$

$$\begin{aligned}R13:\ & A\ Rmc\ (B+C)\ \mathit{or}\ A\ Rmh\ (B+C),\\ & [B]=[C] \Rightarrow A\ Rmh\ B,\ A\ Rmh\ C\end{aligned}$$

$$\begin{aligned}R14:\ & (A+B)\ Rmc\ (C+D),\ A\ Rmc\ C,\ [A]=[C],\\ & [B]=[D] \Rightarrow B\ Rmc\ C,\ [A]+[B]=[C]+[D]\end{aligned}$$

$$\begin{aligned}R15:\ & (A+B)\ Rmc\ (C+D),\ [A]+[B]=[C]+[D],\\ & [A]=[C],\ A\ Rmc\ C \Rightarrow [B]=[D],\ B\ Rmc\ D\end{aligned}$$

$$R16:\ A*B,\ C\ Rmc\ D,\ [A]=[C],\ [B]=[D] \Rightarrow (A+C)*(B+D)$$

$$\begin{aligned}R17:\ & (A+B)\ Rmh\ C\ \mathit{or}\ (A+B)\ Rmc\ C,\ [A]=-[B],\\ & A\ Rmh\ B \Rightarrow A\ Rmh\ C\end{aligned}$$

$$R18:\ (A+B)\ Rmc\ 0 \Rightarrow A\ Rmc\ B$$

Now recall the above example, suppose $[A] = -[B]$ and $A\ Rml\ B$, from Rule 7, the result of $[A] + [B]$ would be $[B]$ and ambiguity is removed.

This approach may be used as a complement to the qualitative reasoning method of de Kleer and Brown. It could reduce ambiguity to some extent by using available information on quantitative relations among variables. Its applications in solving a mass collision problem and in fault diagnosis will be described in the following sections.

5.3 Solving the two mass collision problem through qualitative reasoning

The qualitative reasoning method described in the previous section is used here to solve the two mass collision problem which is used in (Raiman 1986) and, therefore, the result presented in this section can be compared with that of Raiman (1986).

5.3.1 The two mass collision problem

The two mass collision problem is shown in Figure 5.1, where two masses with weight M and m coming from opposite directions with close velocities V_i and v_i. It is required to obtain the qualitative values of the velocities of the two masses after collision, i. e. the directions of the two masses, through qualitative reasoning. In (Raiman 1986), it is assumed that M is much larger than m. If it is only known that $M > m$, then no result can be obtained from Raiman's method since the relation ">" (greater than) is not reflected by the operators he defined. However, this could be solved by the approach presented here.

5.3.2 Qualitative reasoning about the two mass collision problem

Qualitative constraints

From momentum and energy conservations, the following equations can be obtained,

$$MV_i + mv_i = MV_f + mv_f \tag{5.1}$$

$$\frac{MV_i^2}{2} + \frac{mv_i^2}{2} = \frac{MV_f^2}{2} + \frac{mv_f^2}{2} \tag{5.2}$$

where M, V_i, and V_f are the mass, initial and final velocities for the first object respectively, and m, v_i, and v_f are those for the second object respectively.

From Eq(5.1) and Eq(5.2), the following equation can be obtained

$$V_i + V_f = v_i + v_f \tag{5.3}$$

Since it is assumed that the two masses have the same initial velocity, therefore

$$v_f = 2V_i + V_f \tag{5.4}$$

From the above equations, the following constraints can be obtained.

$$(MV_i + mv_i)\ Rmc\ (MV_f + mv_f) \tag{5.5}$$

$$[MV_i + mv_i] = [MV_f + mv_f] \tag{5.6}$$

$$(V_i + V_f)\ Rmc\ (v_i + v_f) \tag{5.7}$$

$$[V_i + V_f] = [v_i + v_f] \tag{5.8}$$

$$(MV_i^2 + mv_i^2)\ Rmc\ (MV_f^2 + mv_f^2) \tag{5.9}$$

$$[v_f] = [2V_i + V_f] \tag{5.10}$$

$$v_f\ Rmc\ (2V_i + V_f) \tag{5.11}$$

The initial conditions of the problem are given by the following constraints.

$$[V_i] = + \tag{5.12}$$

$$[v_i] = - \tag{5.13}$$

$$V_i \; Rmc \; v_i \tag{5.14}$$

$$M \; Rmh \; m \tag{5.15}$$

$$\frac{1}{M} \; Rml \; \frac{1}{m} \tag{5.16}$$

$$[M] = + \tag{5.17}$$

$$[m] = + \tag{5.18}$$

Qualitative reasoning

Applying $R10$ to Eq(5.14) and Eq(5.15), the following equation can be obtained.

$$MV_i \; Rmh \; mv_i \tag{5.19}$$

From $R7$ and Eq(5.19), the following can be obtained.

$$[MV_i + mv_i] = [MV_i] = + \tag{5.20}$$

The qualitative value of V_f can be either +, 0, or −, three hypotheses are generated.

(a). $[V_f] = +$

In this case,

$$[V_i + V_f] = + \tag{5.21}$$

Applying $R9$ to Eq(5.8), Eq(5.13), and Eq(5.21), the following can be obtained.

$$[v_f] = [V_i + V_f] = + \tag{5.22}$$

$$v_f \; Rmh \; v_i \tag{5.23}$$

Applying $R13$ to Eq(5.11), Eq(5.12), and hypothesis (a) gives

$$v_f \; Rmh \; 2V_i \tag{5.24}$$

From Eq(5.17), Eq(5.18), Eq(5.22), and hypothesis (a), the following can be obtained.

$$[MV_f] = + \tag{5.25}$$

$$[mv_f] = + \tag{5.26}$$

Applying $R13$ to Eq(5.5), Eq(5.25), and Eq(5.26) gives

$$(MV_i + mv_i) \; Rmh \; MV_f \tag{5.27}$$

Eq(5.12), Eq(5.13), Eq(5.17), and Eq(5.18) give

$$[MV_i] = -[mv_i] \tag{5.28}$$

Applying $R17$ to Eq(5.27), Eq(5.28), and Eq(5.19) gives

$$MV_i \; Rmh \; MV_f \tag{5.29}$$

from which the following can also be obtained

$$V_i \; Rmh \; V_f \tag{5.30}$$

$$V_f \; Rml \; V_i \tag{5.31}$$

(b).$[V_f] = -$

From Eq(5.6) and Eq(5.20) it follows that

$$[MV_f + mv_f] = + \tag{5.32}$$

Eq(5.17) and hypothesis (b) give

$$[MV_f] = - \tag{5.33}$$

Applying $R9$ to Eq(5.32) and Eq(5.33) gives

$$[mv_f] = + \tag{5.34}$$

$$mv_f \ Rmh \ MV_f \tag{5.35}$$

Eq(5.34) and Eq(5.18) imply that

$$[v_f] = + \tag{5.36}$$

$R1$ and Eq(5.16) give

$$\frac{1}{m} \ Rmh \ \frac{1}{M} \tag{5.37}$$

Applying $R11$ to Eq(5.35) and Eq(5.37) gives

$$v_f \ Rmh \ V_f \tag{5.38}$$

Subhypothesis 1: $v_f \ Rml \ v_i$

From $R7$, we have

$$[v_i + v_f] = [v_i] = - \tag{5.39}$$

From Eq(5.8)

$$[V_i + V_f] = [v_i + v_f] = - \tag{5.40}$$

Applying $R9$ to Eq(5.40) and Eq(5.12) gives

$$V_f \ Rmh \ V_i \tag{5.41}$$

Applying $R4$ to Eq(5.40) and Eq(5.12) gives

$$v_f \ Rmh \ V_i \tag{5.42}$$

Applying $R5$ to Eq(5.42) and Eq(5.14) gives

$$v_f \ Rmh \ v_i$$

which contradicts the subhypothesis and, therefore, the subhypothesis is false.

Subhypothesis 2, $v_f \ Rmc \ v_i$

From $R8$, Eq(5.13), Eq(5.36), Eq(5.7), and Eq(5.8)

$$[V_i + V_f] = [v_i + v_f] = 0 \tag{5.43}$$

$$(V_i + V_f) \ Rmc \ (v_i + v_f) \ Rmc \ 0 \tag{5.44}$$

Applying $R18$ to Eq(5.44) gives

$$V_i \ Rmc \ V_f$$

Applying $R2$ to the above equation gives

$$V_f \ Rmc \ V_i \tag{5.45}$$

Applying $R4$ to Eq(5.45) and Eq(5.14) gives

$$V_f \ Rmc \ v_i$$

Applying $R4$ to the above equation and subhypothesis (2) gives

$$V_f \ Rmc \ v_f$$

which contradicts Eq(5.38) and, hence, this subhypothesis is false. Therefore,

$$v_f \ Rmh \ V_i \tag{5.46}$$

Applying $R5$ to Eq(5.46) and Eq(5.14) gives

$$v_f \ Rmh \ v_i$$

From $R7$ and the above equation

$$[v_i + v_f] = [v_f] = +$$

Eq(5.8) becomes

$$[V_i + V_f] = [v_i + v_f] = +$$

Applying $R9$ to hypothesis (b) and the above equation gives

$$V_i \; Rmh \; V_f$$

Applying $R1$ to the above equation gives

$$V_f \; Rml \; V_i \tag{5.47}$$

(c) $[V_f] = 0$.

Eq(5.10) becomes

$$[v_f] = [2v_i] = [v_i] = + \tag{5.48}$$

$$v_f \; Rmc \; 2v_i \tag{5.49}$$

To summarise, the result is

$$\begin{array}{lll}
(1) & [V_f] = +, & V_f \; Rml \; V_i \\
 & [v_f] = +, & v_f \; Rmh \; 2V_i \\
 & & \\
(2) & [V_f] = -, & V_f \; Rml \; V_i \\
 & [v_f] = +, & v_f \; Rmh \; V_i \\
 & & \\
(3) & [V_f] = 0 & \\
 & [v_f] = +, & v_f \; Rmc \; 2V_i
\end{array}$$

5.3.3 Comparison with analytical solution

The analytical solution of the two mass collision problem is

$$V_f = \frac{M - 3m}{M + m} V_i \tag{5.50}$$

$$v_f = \frac{3M - m}{M + m} V_i \tag{5.51}$$

There exist three possible situations.

(1). $M > 3m$. In this case,

$$V_f > 0, \quad v_f > 0, \quad v_f > 2V_i.$$

(2). $M = 3m$. In this case,

$$V_f = 0, \quad v_f > 0, \quad v_f = 2V_i.$$

(3). $M < 3m$. In this case,

$$V_f < 0, \quad v_f > 0, \quad v_f > V_i.$$

It can be seen that the qualitative reasoning described above gives the correct solution.

5.4 Fault diagnosis of the mixing process

5.4.1 Qualitative modelling of the mixing process

The qualitative model is in the form of a set of confluences which are derived from the quantitative model of the mixing process. The dynamic model of the mixing process, which is developed in Chapter 3, is listed below:

$$A_1 \frac{dH_1}{dt} = Q_c + Q_h - Q_{o1} \tag{5.52}$$

$$A_2 \frac{dH_2}{dt} = Q_{o1} - Q_{o2} \tag{5.53}$$

$$A_1 H_1 \frac{dT_1}{dt} = Q_c(T_c - T_h) + Q_h(T_h - T_1) \tag{5.54}$$

$$A_2 H_2 \frac{dT_2}{dt} = Q_{o1}(T_1 - T_2) \tag{5.55}$$

$$Q_{o1} = K_1 \sqrt{H_1 - H_2} \tag{5.56}$$

$$Q_{o2} = K_2 \sqrt{H_2} \tag{5.57}$$

The parameters and variables in the model are defined in Chapter 3.

One way to derive the confluences is to compare the dynamic model at present state with that at a previous steady state. Compare Eq(5.52) at time $t2$ with that at time $t1$, we have

$$A_1 \frac{dH_1}{dt} \mid_{t2} - A_1 \frac{dH_1}{dt} \mid_{t1} = Q_c \mid_{t2} - Q_c \mid_{t1} + Q_h \mid_{t2} - Q_h \mid_{t1} - Q_{o1} \mid_{t2} + Q_{o1} \mid_{t1} \tag{5.58}$$

Taking the qualitative values of the two sides of Eq(5.58), we have

$$\begin{aligned} [A_1 \tfrac{dH_1}{dt} \mid_{t2} - A_1 \tfrac{dH_1}{dt} \mid_{t1}] &= [Q_c \mid_{t2} - Q_c \mid_{t1}] + [Q_h \mid_{t2} - Q_h \mid_{t1}] - [Q_{o1} \mid_{t2} - Q_{o1} \mid_{t1}] \\ &= \delta_{1,2} Q_c + \delta_{1,2} Q_h - \delta_{1,2} Q_{o1} \end{aligned} \tag{5.59}$$

where $\delta_{1,2}Q_c, \delta_{1,2}Q_h$ and $\delta_{1,2}Q_{o1}$ are the qualitative values of the increments of Q_c, Q_h and Q_{o1} over the time interval $[t1, t2]$ respectively.

Suppose the system is steady at time $t1$, then Eq(5.59) becomes

$$[\frac{dH_1}{dt} \mid_{t2}] = \delta_{1,2} Q_c + \delta_{1,2} Q_h - \delta_{1,2} Q_{o1} \tag{5.60}$$

Eq(5.60) is the confluence for predicting the qualitative value of $\frac{dH_1}{dt}$ at time $t2$. Applying the same procedure to Eqs (5.53) to (5.57), gives

$$[\frac{dH_2}{dt} \mid_{t2}] = \delta_{1,2} Q_{o1} - \delta_{1,2} Q_{o2} \tag{5.61}$$

$$[\frac{dT_1}{dt} \mid_{t2}] = \delta_{1,2} Q_h - \delta_{1,2} Q_c - \delta_{1,2} T_1 \tag{5.62}$$

$$[\frac{dT_2}{dt}\mid_{t2}] = [T_1 - T_2]\delta_{1,2}Q_{o1} + \delta_{1,2}T_1 - \delta_{1,2}T_2 \tag{5.63}$$

$$\delta_{1,2}Q_{o1} = \delta_{1,2}(H_1 - H_2) \tag{5.64}$$

$$\delta_{1,2}Q_{o2} = \delta_{1,2}H_2 \tag{5.65}$$

Eqs(5.60) to (5.65) are the set of confluences which describe the qualitative behaviour of the mixing process. Since these confluences are formally derived from the dynamic model, they are consistent with the dynamic model.

It can be seen that the qualitative model is simpler than the quantitative one. The parameters A_1, A_2, K_1, and K_2 do not appear in the qualitative model, and therefore, the inaccuracies in these parameters will not affect the qualitative model. Compared with the quantitative model, the qualitative one is more robust to slight inaccuracies in measurements or system parameters.

In Eq(5.60), if hand valve 1 is working correctly, $\delta_{1,2}Q_{o1}$ is determined by the difference of H_1 and H_2, and H_1 is determined by Q_c and Q_h. So, $\delta_{1,2}Q_{o1}$ is the feedback effect of $\delta_{1,2}Q_c + \delta_{1,2}Q_h$, and it will have the same sign as $\delta_{1,2}Q_c + \delta_{1,2}Q_h$. This results in ambiguity. Here, to solve this ambiguity, we adopt the same heuristic used by Oyeleye and Kramer (1988). The heuristic is that "an effect cannot compensate for its own cause". Thus,

$$\delta_{1,2}Q_{o1}\ Rml\ \delta_{1,2}Q_c + \delta_{1,2}Q_h \tag{5.66}$$

Applying Rule 7 to Eq(5.66) and (5.60), gives

$$[\frac{dH_1}{dt}\mid_{t2}] = \delta_{1,2}Q_c + \delta_{1,2}Q_h \tag{5.67}$$

Eq(5.67) is used instead of Eq(5.60) when hand valve 1 is working normally. Similarly, if hand valve 2 is working normally, Eq(5.61) can be reduced to

$$[\frac{dH_2}{dt}\mid_{t2}] = \delta_{1,2}Q_{o1} \tag{5.68}$$

Eq(5.62) and Eq(5.63) can be reduced to

$$[\frac{dT_1}{dt}|_{t2}] = \delta_{1,2}Q_h - \delta_{1,2}Q_c \tag{5.69}$$

$$[\frac{dT_2}{dt}|_{t2}] = [T_1 - T_2]\delta_{1,2}Q_{o1} + \delta_{1,2}T_1 \tag{5.70}$$

In the above equations, Q_c and Q_h are not measured. Under normal operating conditions, they should change in the same directions as the corresponding controller outputs which are known. Therefore, in normal conditions, $\delta_{1,2}Q_c$ and $\delta_{1,2}Q_h$ are replaced by $\delta_{1,2}I_c$, where I_c is the controller output to the cold water control valve, and $\delta_{1,2}I_h$, where I_h is the controller output to the hot water control valve, respectively.

5.4.2 Fault detection and diagnosis

Fault detection

Fault detection and diagnosis is based on the qualitative model of the mixing process. The qualitative model provides a set of constraints for the process which should not be violated if there is no fault in the system. The qualitative values of $\frac{dH_1}{dt}$, $\frac{dH_2}{dt}$, $\frac{dT_1}{dt}$, and $\frac{dT_2}{dt}$, i. e. the changing directions of H_1, H_2, T_1, and T_2 respectively, can be calculated from the qualitative model and are compared with the on-line measurements of H_1, H_2, T_1, and T_2 respectively. If the predicted values agree with the actual measurements, there is no fault in the process. Otherwise, it indicates that a fault occurs in the process. Once the presence of a fault is detected, the diagnosis system begins to determine the details of the associated fault.

To reduce the effects of measurement noise, when the predicted behaviour does not agree with the actual measurements, several sets of additional measurements are collected to check model consistency. If in the majority of the cases the model is violated, then there is a fault in the process. Otherwise, the system is still considered to be at a normal condition.

It will waste computer time if the calculations of the expected changing directions are continued regardless whether the measurements are normal or not. To avoid this, an enable condition, which comprises a set of constraint values for the measurements, is defined for the fault detection. If the enable condition is not satisfied, i. e. all the measurements are within their constraint values, the process is considered to be at a normal condition and it is not necessary to calculate the changing directions. It is only when the enable condition is satisfied, that the diagnosis system begins to calculate the expected changing directions from the qualitative model.

Fault diagnosis

Fault diagnosis is performed based on the qualitative model of the mixing process. The model is used to generate the expected behaviour under certain failure hypotheses. Diagnosis is performed by the "hypothesis-test" strategy which contains a procedure of hypothesis generation, simulation and comparison. The procedure is as follows: first, generate a hypothesis based on a particular failure, then simulate the behaviour of the process under this failure. The expected behaviour is compared with the actual measurements, if they agree, this hypothesis is retained. Continuously perform this procedure until all the generated hypotheses have been tested. If no hypothesis is retained, it is an unsuccessful diagnosis. The retained hypotheses are the possible faults.

It will be inefficient when the process being diagnosed contains a large number of components, since the more components it contains, the more hypotheses it will generate. To improve efficiency, the process being diagnosed is decomposed into several subsystems such that the number of components in each subsystem is limited.

The mixing process is divided into two subsystems. The first subsystem includes the hot and cold water control valves, tank 1, and the associated sensors. The second subsystem includes hand valve 1, hand valve 2, tank 2 and the associated sensors.

The possible faults that may occur are considered to be: sensor failures, hot and cold water control valve failures, hand valve 1 and hand valve 2 blocked, and controller failure. Since the parameters, and the inputs and outputs of the controller are known exactly, it is not necessary to derive a qualitative model for the controller to replace the quantitative one. Controller failure is diagnosed by checking the consistency between its inputs and outputs. Sensor failure is diagnosed differently from the failures of other components. Since it is not straight forward to predict the output of a failed sensor, sensor failure is diagnosed from heuristic considerations. These heuristics comprise previous experience on sensor failures and some general knowledge about sensors. During previous operation of the mixing process, the level sensor of tank 2 failed several times. When it failed, its output was fixed at a certain value. Later it is found that this is due to the blockage of the conduit connecting the level sensor and the tank. This gives a heuristic that when a sensor's output is fixed at a certain value, but where other sensor outputs which can directly or indirectly reflect the same measured variable are changing, then the sensor whose output is fixed fails. Another heuristic is that since sensor readings reflect associated process variables and if the process variable changes continuously, the sensor readings should also change continuously, i. e. the change between two successive samples is limited. In the mixing process, the measured variables have large time constants, especially the level variable, and so any abrupt changes in sensor readings reflect sensor failure. From previous experience, when the wires connecting sensors and the computer are broken, the data collected by the computer will change randomly.

To simulate the behaviour of the system under a particular failure, the effect of this failure on the system's model should be characterised. The effects of failures are represented by the deviations of certain process variables and, hence, the qualitative model can be used to simulate the process under normal or faulty conditions.

When hand valve 1 is blocked or partially blocked, the water flow between tank 1 and tank 2 will decrease, thus

$$\delta_{1,2}Q_{o1} = - \tag{5.71}$$

Similarly, if hand valve 2 is blocked or partially blocked,

$$\delta_{1,2}Q_{o2} = - \tag{5.72}$$

If the cold water control valve fails, its average output flow rate will be either higher or lower than the normal one. If it is lower, the level in tank 1 will decrease and subsequently cause the level in tank 2 to decrease. Since level 2 is being controlled, the decrease in level 2 will cause the input to the cold water control valve to increase. Similarly, if the output flow rate of the cold water control valve is higher than the normal value, the input to the cold water control valve will decrease. Therefore, when the cold water control valve fails

$$\delta_{1,2}Q_c = -\delta_{1,2}I_c \tag{5.73}$$

Similarly, when the hot water control valve fails

$$\delta_{1,2}Q_h = -\delta_{1,2}I_h \tag{5.74}$$

When a fault is detected, the hypothesis generator generates an hypothesis based on the observed symptom which comprises the information on which measurements suggest different behaviour from the predictions.

From Equations (5.67) to (5.70), it can be seen that T_1 and Q_{o1} appear in the models of both subsystems. Q_{o1} is determined by Equation (5.56). So, the failure of hand valve 1, the failures of both level sensors, and the failure of the temperature sensor of tank 1 will affect both subsystems. These failures are arranged together to form a common list, while the other failures are arranged into another two lists corresponding to the subsystem to which they belong. The arrangement of candidate lists is shown in Figure 5.2.

If only the model of the first subsystem is violated, then the hypothesis is generated from list 1, whereas if only the model of the second subsystem is violated, the hypothesis is generated from list 2. If the models of both subsystems are violated, then the hypothesis is initially generated from the common list. If all the candidates in the common list have been tried, and the models of both subsystems are still violated, then the hypothesis is generated from list 1 and list 2. The hypothesis is generated by heuristic rules which are in the following form:

IF Symptom **THEN** Hypothesis

where the symptom includes the pattern of abnormal measurements, i. e. which particular measurements significantly deviate from their steady state values, and the pattern of contradictions, i. e. which variable's behaviour is different from its prediction. For example, if only the temperature measurements are abnormal, then the hypothesis is generated from the set of failures which can affect the temperature control loop, i. e. temperature sensor failures and hot water control valve failure.

Since sensor failures will affect the qualitative simulation, they are arranged to be at the top of candidate lists such that they can be hypothesised prior to other component failures. Therefore, when a fault is detected, the diagnosis system first tries to find out if the sensors are working normally. If the sensors are working normally, then the measurements are reliable, and thus, the qualitative simulation will also be reliable.

5.4.3 Performance of the diagnosis system

The fault diagnosis system has been successfully applied to the mixing process. During the experiments, all the faults mentioned above were separately initiated, and they were diagnosed very successfully. Table 5.4 shows the result of the experiment

Table 5.4: Performance under a single failure

fault initiated	no. of successful diagnosis
temp. sensor 1 fail	3
temp. sensor 2 fail	4
level sensor 1 fail	3
level sensor 2 fail	4
hand valve 1 blocked	5
hand valve 2 blocked	5
hot water control valve fail	5
cold water control valve fail	5

in which every fault was initiated five times. It can be seen that the performance is very satisfactory.

The result of the experiment shows that the performance of the diagnosis system subjected to sensor failures is not as good as that subjected to other component failures. This is due to the fact that measurements in the mixing process are not abundant and, therefore, sensor failures are diagnosed mainly by detecting abrupt changes in sensor readings. Sometimes, when a sensor fails, the change in its reading is not abrupt and, therefore, this fault is missed. In a sensor rich environment, sensor failures are easier to diagnose (Scarl, Jamieson, Delaune 1987).

During simulation studies and experiments, it has been found that the diagnosis system can diagnose partial blockage of hand valves. The simulation result shows that this fault can still be diagnosed when the hand valves are only 20% blocked.

Experiments have been conducted when several faults were initiated simultaneously. Since each different fault takes a different time to affect the control system, only the fault with a quick effect was diagnosed.

5.5 Fault diagnosis of a CSTR system

A qualitative modelling based diagnosis system is also developed for the CSTR system in a similar way as that described in the previous section.

5.5.1 Qualitative modelling of the CSTR system

The qualitative model of the CSTR system is derived from its dynamic model, which is developed in the previous chapter and is listed below.

$$A\frac{dH}{dt} = Q_1 + Q_2 - Q_3 \tag{5.75}$$

$$AH\frac{dC_a}{dt} = Q_1(C_{a0} - C_a) - r_a AH \tag{5.76}$$

$$AH\frac{dC_b}{dt} = r_a AH - C_b Q_1 \tag{5.77}$$

$$AHB_2\frac{dT}{dt} = B_1 Q_1(T_1 - T) - B_2 Q_2(T - T_2) + H_r r_a \tag{5.78}$$

$$B_1 = C_{a0}\rho C + (1 - C_{a0})\rho_0 C_0 \tag{5.79}$$

$$B_2 = \rho C(C_a + C_b) + (1 - C_a - C_b)\rho_0 C_0 \tag{5.80}$$

$$r_a = K_r C_a^n \qquad (n > 0) \tag{5.81}$$

$$K_r = a_r e^{-b_r/T} \tag{5.82}$$

$$Q_2 = K_2 A_2 \sqrt{P} \tag{5.83}$$

$$Q_4 = K_4 A_4 \sqrt{P} \tag{5.84}$$

$$Q_3 = Q_2 + Q_4 \tag{5.85}$$

$$P = P_0 + \Delta P \tag{5.86}$$

$$P_0 = H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \tag{5.87}$$

$$Q_5 = K_5 A_5 \sqrt{P_5} \tag{5.88}$$

$$T_2 = \frac{C_0\rho_0 Q_5 T_5 + Q_2 T[C\rho(C_a + C_b) + C_0\rho_0(1 - C_a - C_b)]}{C_0\rho_0 Q_5 + Q_2[C\rho(C_a + C_b) + C_0\rho_0(1 - C_a - C_b)]} \tag{5.89}$$

The parameters and variables in the model are defined in the previous chapter.

It is assumed that the process is operating at a steady state prior to the occurrence of a fault. Therefore, the qualitative model for the CSTR system can be derived based on its steady state model. Under this assumption, from Eq(5.75) and Eq(5.84) to Eq(5.87) the following equation can be obtained.

$$\begin{aligned} Q_1 &= Q_4 \\ &= K_4 A_4 \sqrt{P} \\ &= K_4 A_4 \sqrt{H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] + \Delta P} \end{aligned} \tag{5.90}$$

In steady state, Eq(5.76) to Eq(5.78) become

$$Q_1(C_{a0} - C_a) = r_a A H \tag{5.91}$$

$$Q_1 C_b = r_a A H \tag{5.92}$$

$$(B_1 Q_1 + B_2 Q_2)T = B_1 Q_1 T_1 + B_2 Q_2 T_2 + H_r r_a \tag{5.93}$$

The qualitative model is obtained by first differentiating and then taking qualitative values of the two sides of the quantitative equations as used in (de Kleer and Brown 1984). To simplify the qualitative model, several practical assumptions are also made.

Differentiating the two sides of Eq(5.90) gives

$$\begin{aligned} \tfrac{dQ_1}{dt} = \; & K_4\{H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] + \Delta P\}\tfrac{dA_4}{dt} \\ & + K_4 A_4\{[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0]\tfrac{dH}{dt} \\ & + H(\rho - \rho_0)(\tfrac{dC_a}{dt} + \tfrac{dC_b}{dt}) + \tfrac{d\Delta P}{dt}\} \end{aligned}$$

In the above equation, it is assumed that the changes in C_a and C_b cannot significantly affect the average density of the content in the reactor and, therefore, cannot significantly affect the pressure at the bottom of the reactor, then the above equation can be simplified as

$$\begin{aligned}
&K_4A_4[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0]\frac{dH}{dt} \\
&= \frac{dQ_1}{dt} - K_4\{H[(C_a + C_b)\rho + (1 - C_a - C_b)\rho_0] \\
&+\Delta P\}\frac{dA_4}{dt} - K_4A_4\frac{d\Delta P}{dt}
\end{aligned} \tag{5.94}$$

Taking the qualitative value of the two sides of Eq(5.94) and using δX to denote $[\frac{dX}{dt}]$, Eq(5.94) becomes

$$\delta H = \delta Q_1 - \delta A_4 - \delta \Delta P \tag{5.95}$$

Similarly, differentiating and taking the qualitative value of the two sides of Eq(5.91) and Eq(5.92) gives

$$\delta C_a = \delta Q_1 + \delta C_{a0} - \delta H - \delta T \tag{5.96}$$

$$\delta C_b = \delta H + \delta T + \delta C_a - \delta Q_1 \tag{5.97}$$

In Eq(5.93), it is assumed that the changes in C_{a0}, C_a, and C_b will not significantly affect the densities and specific heats of the input reactant and the content in the reactor, therefore, B_1 and B_2 in Eq(5.93) can approximately be treated as constants. Then, differentiating the two sides of Eq(5.93) gives

$$\begin{aligned}
&(B_1Q_1 + B_2Q_2)\tfrac{dT}{dt} + TB_1\tfrac{dQ_1}{dt} + TB_2\tfrac{dQ_2}{dt} \\
&= B_1Q_1\tfrac{dT_1}{dt} + B_1T_1\tfrac{dQ_1}{dt} + B_2Q_2\tfrac{dT_2}{dt} + B_2T_2\tfrac{dQ_2}{dt} \\
&+H_r[K_rnC_a^{n-1}\tfrac{dC_a}{dt} + \tfrac{C_a^n a_r b_r e^{-b_r/T}}{T^2}\tfrac{dT}{dt}]
\end{aligned}$$

The above equation can be re-formulated as

$$(B_1Q_1 + B_2Q_2 - \frac{H_rC_a^n a_r b_r e^{-b_r/T}}{T^2})\frac{dT}{dt}$$

$$= B_1 Q_1 \frac{dT_1}{dt} + B_2 Q_2 \frac{dT_2}{dt} - B_1(T - T_1)\frac{dQ_1}{dt} - B_2(T - T_2)\frac{dQ_2}{dt} + H_r K_r n C_a^{n-1} \frac{dC_a}{dt} \tag{5.98}$$

From the parameter values provided in the previous chapter, the value of the expression in the round bracket of the left hand side of Eq(5.98) is positive, therefore, taking qualitative values of the two sides of Eq(5.98) gives

$$\delta T = \delta T_1 + \delta T_2 - \delta Q_1 - \delta Q_2 + \delta C_a \tag{5.99}$$

Similarly, the following can be obtained from Eq(5.83) to Eq(5.89).

$$\delta Q_2 = \delta A_2 + \delta P \tag{5.100}$$

$$\delta Q_4 = \delta A_4 + \delta P \tag{5.101}$$

$$\delta Q_3 = \delta Q_2 + \delta Q_4 \tag{5.102}$$

$$\delta P = \delta P_0 + \delta \Delta P \tag{5.103}$$

$$\delta P_0 = \delta H \tag{5.104}$$

$$\delta Q_5 = \delta A_5 + \delta P_5 \tag{5.105}$$

$$\delta T_2 = \delta T5 + \delta T + \delta Q_2 - \delta Q_5 \tag{5.106}$$

So far the qualitative model for the CSTR system has been developed. To simulate the effect of a fault, the fault should be represented as a deviation in the corresponding process variable as described in the previous section. The representations of the possible faults (except sensor failures) in the CSTR system are given in Table 5.5.

Table 5.5: Representations of faults

Faults	Representations
Pipe 1 is blocked	$\delta Q_1 = -$
External feed reactant flow high	$\delta Q_1 = +$
Pipe 2 or 3 is blocked or pump fails	$\delta \Delta P = -$
External feed reactant temp. high	$\delta T_1 = +$
External feed reactant temp. low	$\delta T_1 = -$
Pipe 10 or 11 is blocked or control valve 1 fails low	$\delta A_4 = -$
Control valve 2 fails high	$\delta A_5 = +$
Pipe 7, 8, or 9 is blocked or control valve 2 fails low	$\delta A_5 = -$
Control valve 1 fails high	$\delta A_4 = +$
Pipe 4, 5, or 6 is blocked or control valve 3 fails low	$\delta A_2 = -$
Control valve 3 fails high	$\delta A_2 = +$
External feed reactant concentration too low	$\delta C_{a0} = -$

5.5.2 Fault detection and diagnosis

The on-line fault diagnosis system for the CSTR system is similar to that for the mixing process described in the previous section. An enable condition, which consists of several constraints on the measurements, is defined. Only when this enable condition is satisfied, does the diagnosis system begin to detect and diagnose faults. Fault detection is performed by predicting the behaviour of the process under normal operating conditions and comparing this with the actual measured behaviour. A fault is detected if the predicted behaviour differs from the actual one.

To improve efficiency, the CSTR system is decomposed into two subsystems. The first subsystem includes pipe 1, reactor, pump, pipes 2, 3, 10, 11, valve 1, and sensors associated with these components. The rest form the second subsystem. Fault diagnosis is performed through the "hypothesis-test" strategy. If a hypothesis is a sensor failure, then it is discriminated by heuristic rules. Other hypotheses are discriminated through qualitative simulation.

5.6 Conclusions

Process fault diagnosis based on qualitative modelling is investigated in this chapter. It is demonstrated that qualitative reasoning depends less on accurate process model parameters and accurate measurements and, consequently, the result obtained from qualitative reasoning is less accurate than that of quantitative reasoning. However, for the purpose of fault diagnosis, accurate reasoning is generally not needed and, sometimes, is difficult to implement. Ambiguity is a problem associated with qualitative reasoning. It is demonstrated in this chapter that ambiguity could be reduced by taking account of certain available quantitative information. The model of a process can be greatly simplified if only the signs (+, 0, −) of process variables are concerned. If the order of magnitude information is used in qualitative reasoning, then only limited simplification is allowed to preserve the order of magnitude information. There is a conflict between model simplification and obtaining a less ambiguous result.

Based on qualitative modelling, process fault diagnosis can be performed through the "hypothesis-test" strategy. Since the behaviour of the process under certain failures, such as sensor failures, may not be predicted efficiently through qualitative simulation, an approach combining qualitative reasoning and heuristic reasoning should be used. Through decomposing the system being diagnosed into several subsystems, diagnosis can be rapidly focused in a small region.

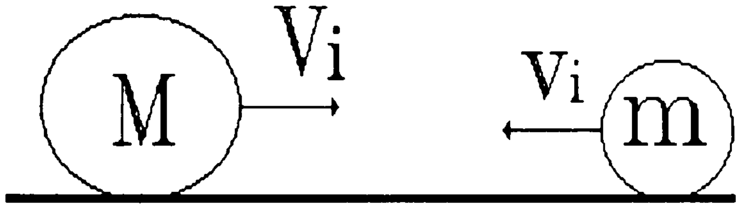
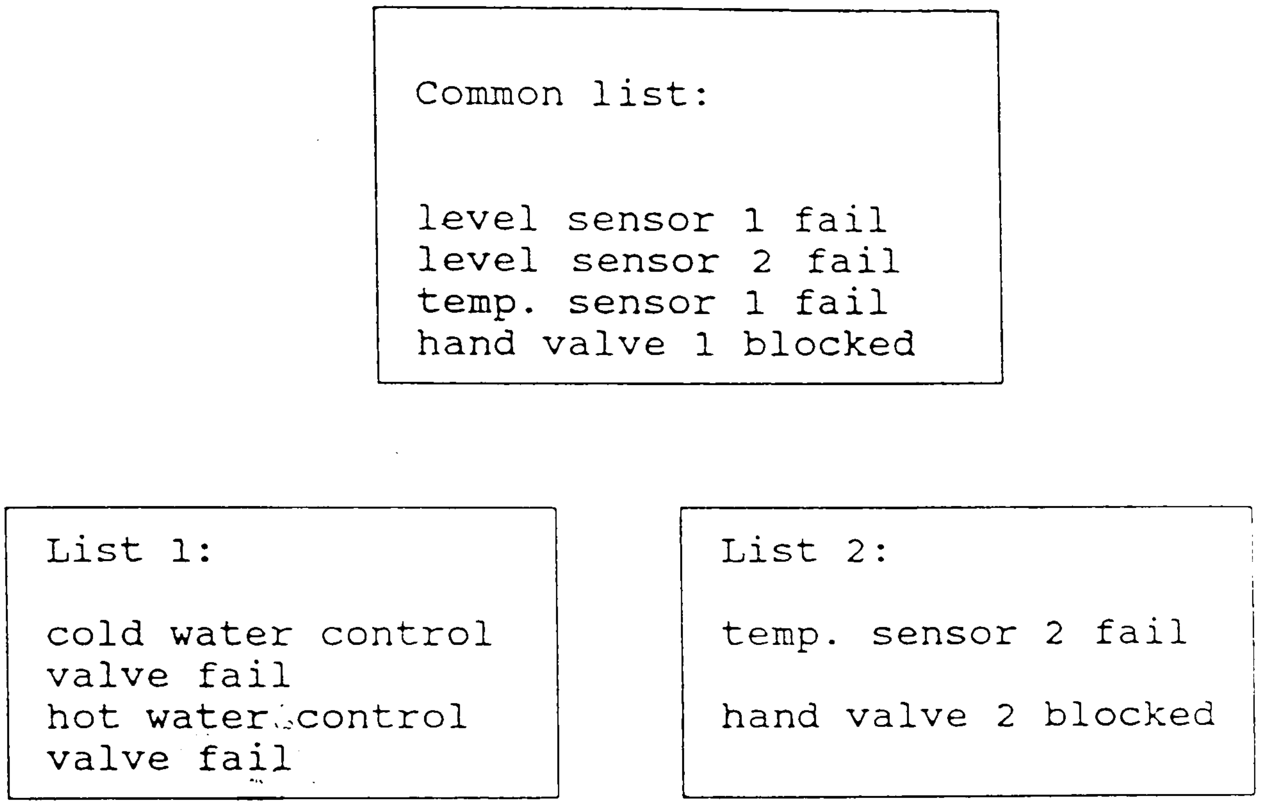


Fig.5.1 Two mass collision problem



Common list:

level sensor 1 fail
level sensor 2 fail
temp. sensor 1 fail
hand valve 1 blocked

List 1:

cold water control valve fail
hot water control valve fail

List 2:

temp. sensor 2 fail

hand valve 2 blocked

Figure 5.2 Candidate lists

Chapter 6

Qualitative simulation based fault diagnosis with self-reasoning facility

6.1 Introduction

With the increasing complexity of expert systems, it would be desirable to design a system that can reason its own behaviour and thus find its own defects and improve its performance by correcting these defects. That is it can learn from past experience. Since diagnosis is a dominant application area of expert systems, the ability of learning would be a desirable property for a fault diagnosis system and, recently, several fault diagnosis systems with a learning property have been reported (Pazzani 1986, 1987, Rich and Venkatasubramanian 1989). They are called failure-driven learning diagnosis systems because learning is initiated when a failure occurs in diagnosis.

In these systems, fault diagnosis is based on a set of heuristic rules, which are believed to give efficient diagnosis. These heuristic rules are in the form:

IF Symptoms **THEN** Fault.

Since the heuristic rules may not be perfect, a failure may occur during diagnosis in that the hypothesis proposed by a rule is incorrect. Once such a failure has occurred, the heuristic rule generating the wrong hypothesis is modified and a new rule is generated. The task of learning is carried out based on a deep model of the system being diagnosed. From this deep model, the other effects of the proposed fault, which are not included in the condition part of the failed heuristic rule, can be obtained and another fault, which may cause the same symptom as the condition part of the failed heuristic rule, can also be obtained. The failed heuristic rule is modified by including additional features in its condition part, which are obtained from reasoning through the deep model, such that its applicability is limited and will not be employed in future similar situations. A new heuristic rule corresponding to the newly discovered fault from reasoning through the deep model is added.

Generally, in the diagnosis of a complex system, such as a nuclear reactor (Nelson 1982), the diagnosis result is usually obtained by the chaining of a set of rules, and some of the rules are not in the form: **IF** Symptoms **THEN** Fault. Therefore, when a failure occurs in fault diagnosis, it may not be easy to decide which particular rule is responsible for this failure and, hence, the above described method may not be applied in a straight forward manner to the diagnosis of complex systems.

In this Chapter, a self-learning fault diagnosis system, where the task of learning is carried out differently from above, is described. It is based on the fault diagnosis system described in Chapter 5, which diagnoses faults based on a deep qualitative model of the process being monitored. More such qualitative model based fault diagnosis systems have been reported recently (Herbert and Williams 1986, 1987, Oyeleye and Kramer 1988, Waters and Ponton 1989), which demonstrates the popularity of the qualitative model based approach in process fault diagnosis. From this qualitative model, the expected behaviour of the process can be generated and, if it is different from the actual one, then it is perceived that a fault (or faults) occurs in the process. Fault diagnosis is performed by generating a set of hypotheses, each assuming a specific fault occurring, which are tested using heuristic rules or qualitative simulation depending on the nature of a particular hypothesis. The hypotheses assuming sensor failures are discriminated by a set of heuristic rules, while other hypotheses are tested by qualitatively simulating the effect of a particular fault on the process and comparing this with the actual measurements and, depending on whether they match or not, a hypothesis is confirmed or rejected. The threshold values for converting quantitative values to qualitative values and those used in sensor failure diagnosis will affect the performance of the system, and the inappropriate settings of these parameters are considered as a major reason for failures in diagnosis. Once such a failure occurs, the self-learning fault diagnosis system will examine the recorded problem solving history and reason its own behaviour. It will try to find any inappropriate threshold values and give a diagnosis result under new values. The self-learning fault diagnosis system can be viewed as a hierarchical fault diagnosis system where the lower level diagnosis system is an ordinary one as described in the previous chapter and the upper level one can reason the behaviour of the lower level one if it failed to give a correct result.

In the next section, a detailed description of the self-learning diagnosis system is given. Section 6.3 describes the application in the fault diagnosis of the mixing process. A case study is given to illustrate how the self-learning fault diagnosis system works, and this is followed by a description of the performance of the system. The last section contains some concluding remarks.

6.2 Self-learning fault diagnosis

When the self-learning fault diagnosis system fails to give a correct result, it begins to investigate its own behaviour. There are two kinds of such failures: one is that the diagnosis result is wrong, another one is that the system has perceived that a fault (or faults) occurs in the process but no diagnosis result is presented. The reasons for the failures are considered to be: incorrect qualitative models, this could be either that the model developed for the normal operating conditions is incorrect, which could lead to a wrong fault detection, or some of the models developed for various faulty condition are invalid, which could result in a wrong diagnosis; incorrect generation of hypothesis, for example, the generated hypotheses do not include the real fault; and incorrect settings of certain parameters which set the thresholds for converting quantitative values to qualitative values and the thresholds used to diagnose sensor failures. Here the major reason is considered to be inappropriate settings of certain thresholds. These will dramatically affect the diagnosis. Sometimes, if the effect of a malfunction is slight, then certain measurements may be at their thresholds and, therefore, the diagnosis is sensitive to the incremental changes in the plant state. This is referred to as diagnostic instability (Kramer 1987). Shiozaki *et al* (1985) show the superiority of using five-range patterns of abnormality to using three-range patterns of abnormality. In their work, they use SDG (Signed, Directed Graphs) with five-range patterns of abnormality to diagnose chemical plant faults. In the previous SDG approach (Iri, O'Shima, and Matsuyama 1979), the state of a process variable is described by one of the following signs: $+$, 0, and $-$, where $+$ stands for higher than normal, 0 for normal, and $-$ for lower than normal. The problem with this approach is that it is difficult to determine the threshold values, and any inappropriate values can result in a wrong diagnosis. Shiozaki *et al* (1985) modify this approach by using five-range patterns ($+$, $+?$, 0, $-?$, $-$) to describe the states of process variables, where $+?$ and $-?$ indicate the uncertainties between

+ and 0, and − and 0 respectively. It is shown that by such means the possibility of a wrong diagnosis can be reduced. Here a possible range for each threshold value is defined, such that the threshold values can vary within their ranges. Through reasoning its own behaviour, the diagnosis system will find any inappropriate parameters and suggest correct ones. It will then present the diagnosis result under the new parameters. This can also be viewed as failuredriven learning since learning is initiated when the diagnosis system fails to give a correct result.

As pointed out by Hudlická and Lesser (1987), a problem solving system has the following characteristics: 1) complete knowledge of internal system structure; 2) availability of the intermediate problem solving states; 3) large amount of data to process during diagnosis; 4) in some cases, lack of absolute standards for correct behaviour. With the first two properties, it would be desirable to design a self-learning fault diagnosis system which investigates its own behaviour based on its own model.

6.2.1 Model of the fault diagnosis system

The fault diagnosis system contains two parts: fault detection and fault diagnosis. Fault detection is performed by comparing the actual behaviour of the process being diagnosed, which comprises the qualitative increments (increase, steady, or decrease) of certain measured variables over a period, with its prediction, as is illustrated in Figure 6.1.

In Figure 6.1, the controlling input to the process being diagnosed and the resulting on-line measurements are converted into qualitative values by a quantitative to qualitative value converter. A qualitative simulator then simulates the process and predicts the qualitative increments of certain measured variables. These predictions are compared with the qualitative increments converted from on-line measurements. If they are identical, then no fault is identified. If they are different, then the measurements of several successive samples are taken to eliminate the effect of measurement noise. Here NVS (Number of Violated Samples) is used to represent the number of samples in which the actual and predicted qualitative increments are different. If NVS is greater than a pre-defined threshold, N_f, then, and only then, is it perceived that a fault (or faults) occurs in the process. Once such a situation is encountered, the diagnosis system begins to diagnose faults.

The diagnosis methods for sensor failures and other component failures are different, as illustrated in Figure 6.2. If the generated hypothesis indicates a sensor failure, then it is confirmed or denied by a set of heuristics. In these heuristic rules, symptoms are linked by logical operators: **AND** and **OR**, as is illustrated in Figure 6.3 where, if symptom C is presented, or both symptoms A and B are presented, then it is indicated that the sensor has failed. The symptoms are determined by comparing the on-line measurements with pre-defined thresholds. For example, one of the heuristic rules is to check if the increment of a measurement between two successive samples is too high and, if it is, then it indicates sensor failure. The quantitative increment of a measurement between two successive samples is compared with a threshold to determine if it is too high or not.

If the generated hypothesis indicates the failure of other components rather than sensors, then it is discriminated through qualitative simulation as illustrated in Figure 6.4. It can be seen that the diagnosis of the failures of non-sensor components is similar to the fault detection shown in Figure 6.1. The difference is: for fault detection, the qualitative simulator simulates the behaviour of the process under normal conditions; while, for the diagnosis of non-sensor components, the simulator simulates the behaviour under a given hypothesis which is the assumption that some components have failed. In Figure 6.4, the qualitative increments of certain measured variables are compared with their predictions which are calculated through qualitative simulation. This procedure is repeated for all the recorded successive samples. If NVS is less than a pre-defined threshold value, N_d, then the hypothesis is confirmed. If the generated hypothesis is not confirmed, the fault diagnosis system will generate another hypothesis and repeat the above procedure until a fault is diagnosed or all the possible candidates have been tested.

6.2.2 Reasoning the behaviour of the fault diagnosis system

Reasoning the behaviour of the fault diagnosis system can be done by backward tracing through its model. When a failure occurs in diagnosis, an expected output of the system is set, which is propagated backwards through the model of the fault diagnosis system. The threshold values which are responsible for not giving the expected output are then examined to determine the change of which threshold values will give the expected output.

As mentioned previously, any inappropriate threshold values could result in failures in fault diagnosis, and there are two kinds of such failures. One is that the diagnosis result is wrong, and the other is that it is detected that a fault (or faults) occurs in the process but no diagnosis result is presented. The failure may lie in the fault detection part, that is there is actually no fault but it is detected that a fault (or faults) occurs, or the fault diagnosis part, that is the diagnosis result is wrong or no result is given. Since the fault diagnosis starts when it has detected that a fault (or faults) occurs in the process, the self-learning diagnosis system will first examine the fault detection part. It will then try to find out whether there really is a fault in the process being monitored.

Examining the fault detection part

To examine the fault detection part the self-learning fault diagnosis system will carry out backward tracing through the model of this part as shown in Figure 6.1. It will try to deny the fault detection by changing certain threshold values within acceptable ranges. To do this, it will first give an expectation that there is no fault at the output of the fault detection part. Then this expectation is propagated backwards through the model. To deny the fault detection, NVS should be decreased such that it is lower than N_f. The value of NVS is determined by the discrepancies between predicted behaviour and actual behaviour and to reduce NVS, it will then examine which variable's qualitative increment is different from its prediction. Then it will try to change the threshold values, which are related to the conversion of this variable from its quantitative value to a qualitative value and to the qualitative simulation for predicting this variable's qualitative increment, within certain ranges to see if NVS can be decreased below the defined threshold value N_f. If it can, then the fault detection is denied and the new threshold values are recorded. Otherwise, the fault detection cannot be denied and the fault diagnosis part should be examined.

The conversion from a quantitative value to a qualitative value of a variable A is performed by comparing the quantitative value with pre-defined threshold values A_+ and A_- such that

$$[A] = \begin{cases} +, & A > A_+, \\ 0, & A_- \leq A \leq A_+, \\ -, & A < A_-. \end{cases}$$

It can be seen that the qualitative value may change when the threshold values are changed. To reduce NVS, the associated threshold values should be changed such that the predicted and actual qualitative increments will move towards correspondence. For example, if the predicted and actual qualitative increments are $+$ and $-$ respectively, then the associated threshold values should be changed in such a way that the two qualitative increments will move towards 0, while if the predicted and actual increments are + and 0 respectively, then the associated threshold values should be changed so as to either move the predicted value to 0 or move the actual value to +. Since the result of changing the threshold values related to the determination of actual qualitative increment can be easily obtained, these threshold values are changed first such that the actual qualitative increment can be moved towards its prediction. The way to change threshold values is illustrated by the following example. Suppose that it is required to change $[A]$ in the above equation from $-$ to 0, then the threshold value A_- should be reduced to increase the range $[A_-, A_+]$ which corresponds to the qualitative value 0. If by this means NVS can be reduced below its threshold N_f, then the fault detection is denied. Otherwise, the threshold values relating to the calculation of the predicted qualitative increments are changed such that the predicted values will change towards the actual ones. If NVS can be reduced below its threshold N_f, then the fault detection is denied and the new threshold values are recorded. If the fault detection cannot be denied, then it is believed that there is really a fault (or faults) in the process and the fault diagnosis part should be examined.

Examining the fault diagnosis part

A failure in the fault diagnosis part can be in the form that a diagnosis result is wrong or that no diagnosis result is presented. The self-learning diagnosis system will examine the recorded problem solving history. It will examine the generated hypothesis and try to confirm the hypothesis which is denied by the diagnosis system and to deny the wrong diagnosis. This can be summarised as an algorithm:

Step 1. Let the hypothesis be the initially generated hypothesis.

Step 2. If this hypothesis was confirmed by the fault diagnosis system in that it is the diagnosis result, then perform the sub-task: deny hypothesis, if it can be denied the new threshold values will be recorded, then, go to **Step 3**; if this hypothesis was denied by the diagnosis system, then perform the sub-task: confirm hypothesis, if it is confirmed, then record the new threshold values and exit, else, go to **Step 3**.

Step 3. If the hypothesis is the last one in the recorded problem solving history, then exit; else, let the hypothesis be the next generated hypothesis and go to **Step 2**.

Since sensor failures are diagnosed differently from other component failures, the sub-tasks of denying and confirming hypotheses for these particular failures are carried out differently. If the task is to deny a sensor failure, then the self-learning system will trace backwards through the recorded diagnosis history and find out which symptom resulted in this diagnosis. Next, it will examine if this symptom can be eliminated by changing the related threshold values within certain ranges. If it can, then this hypothesis can be denied by changing the related threshold values.

If the task is to confirm sensor failure, then the selflearning diagnosis system will trace through the recorded diagnosis history and examine if some symptoms necessary for confirming sensor failure can be established by changing the related threshold values. If, indeed, it is found that these symptoms can be established by changing certain threshold values, then this hypothesis can be confirmed.

The tasks of confirming or denying other component failures are carried out by backward tracing through the model of the fault diagnosis part as shown in Figure 6.4. Hypothesis confirmation is performed in a similar way as the task of denying fault detection which is described earlier. To confirm a hypothesis, NVS should be reduced such that it is lower than the threshold N_d. This may be achieved by changing the associated threshold values in a similar way, as described previously, to deny fault detection.

To deny a hypothesis, NVS should be increased such that it is not lower than the threshold N_d. The associated threshold values should be changed in such a way that the predicted qualitative increment and the actual one will move in opposite directions to extend their differences, and so that NVS will increase. For example, if the predicted and actual qualitative increments have the value +, then the associated threshold values should be such changed that one of the qualitative increments will move to 0.

6.3 Implementation

6.3.1 Fault diagnosis of a mixing process

The above described self-learning diagnosis techniques have been applied to the fault diagnosis of the pilot scale mixing process. A fault diagnosis system which diagnoses faults based on a qualitative model of the mixing process has been developed and described in the previous chapter. Based on the qualitative model, the qualitative increments of the measured variables are calculated and are compared with the actual measurements. If they do not match, then it is detected that a fault (or faults) occurs, and the diagnosis system begins to diagnose this fault. Within this diagnosis system, the identification of sensor failures is based on heuristic rules while the other system component failures are diagnosed by simulating the process under a hypothesis and comparing the simulated behaviour with the actual one. If the actual behaviour follows the simulated one, then the hypothesis is confirmed.

The performance of the fault diagnosis system is affected by the threshold values which are related with the conversion from quantitative values to qualitative values and the determination of symptoms in the diagnosis of sensor failures. It would be desirable that the fault diagnosis system can reason its own behaviour such that any inappropriate settings of threshold values can be determined and the performance of the system will be improved. A self-learning fault diagnosis system is developed for achieving such a requirement. When the system fails to give a correct result, the self-learning fault diagnosis system will examine its own behaviour and determine any inappropriate threshold values. To do this, a set of ranges in which each threshold value can vary are defined. The threshold values used, together with their ranges are shown in Table 6.1, where $CT1(1)$ to $CT1(17)$ are the currently used threshold values, $VT1$ and $VT2$ are the corresponding maximum and minimum possible values for each threshold. The threshold values with units "*cm*" and "°*C*" are used for level and temperature measurements respectively, while the others are used for outputs of control valves with "%" indicating the percentage of opening. The first nine threshold values are used to convert quantitative increments, in measurements and controller outputs, to their qualitative forms. For example, $CT1(1)$ is used to determine the qualitative increment of temperature in tank 1 as follow:

$$[\Delta T_1] = \begin{cases} +, & \text{if } \Delta T_1 > CT1(1), \\ 0, & \text{if } -CT1(1) \leq \Delta T_1 \leq CT1(1), \\ -, & \text{if } \Delta T_1 < -CT1(1). \end{cases}$$

The other threshold values are used in the diagnosis of sensor failures.

6.3.2 Implementation language

The self-learning fault diagnosis system has been implemented in an expert system shell: **ExTran** (Razzak, Hassan, and Ahmad 1986). The self-learning fault diagnosis system is defined by a main problem, EFD, together with 26 sub-problems. Each sub-problem performs a specified task. Corresponding to each problem, there is a

Table 6.1: The used threshold values and their ranges

	$CT1$	$VT1$	$VT2$
1	0.15°C	0.4°C	0.0°C
2	0.15°C	0.4°C	0.0°C
3	0.1cm	0.3cm	0.0cm
4	0.1cm	0.3cm	0.0cm
5	8.0%	20.0%	0.0%
6	8.0%	20.0%	0.0%
7	2.0%	6.0%	0.0%
8	1.5%	3.0%	0.0%
9	2.0°C	5.0°C	0.0°C
10	4.0cm	6.0cm	2.0cm
11	8.0°C	10.0°C	5.0°C
12	0.1cm	0.4cm	0.08cm
13	0.04cm	0.06cm	0.0cm
14	0.3°C	0.6°C	0.2°C
15	0.05°C	0.1°C	0.0°C
16	3.0°C	7.0°C	2.5°C
17	3.0cm	4.0cm	2.0cm

rule file. The rules in these files can be provided by the designer or induced by **ExTran** from given examples. Since the self-learning diagnosis system reasons its own behaviour from its model, the rules are provided by the designer.

6.3.3 A case study

Since the fault diagnosis system for the mixing process described in Chapter 5 is well tuned, all the threshold values are set appropriately. To test the self-learning diagnosis system, initially it is required to deviate some threshold values from their pre-set values. In this example, we have set the 16*th* threshold value, $CT1(16)$, related to the diagnosis of temperature sensor failure, to 6.0°C. Its previous value was 3.0°C and its range is considered to be $[2.5°C, 7.0°C]$. The corresponding diagnostic rule is:

IF Temperature in tank 2 is at its setpoint
AND
The difference between temperatures in tank 1 and
tank 2 is greater than $CT1(16)$
THEN Temperature sensor in tank 1 has failed

The threshold value was set by entering the conversational mode of the supervisory program. The conversation between the process operator and the computer covering this event is shown in Figure 6.5, where the italics are the operator's reply. After changing this threshold value, the failure of temperature sensor 1 is initiated. The diagnosis result under this inappropriate threshold is "Hot water control valve fail". After being informed that the diagnosis result is wrong, the self-learning diagnosis system begins to examine its own behaviour. It then finds that the 16*th* threshold value is set too high, and if this threshold value is reduced to 5.0°C, the diagnosis result would be "Temperature sensor 1 fail". Figure 6.6 is a copy of the information displayed on the screen. In Figure 6.6, the process is initially operated at its steady state. After time block number 44, a temperature sensor 1 failure, in the form that its output deviated to 35°C instead of the normal value 40°C, is initiated, and a diagnosis result is given after time block number 45. The self-learning fault diagnosis system was informed that the diagnosis result is wrong after time block number 48. This was done by entering the conversational mode of the supervisory program and, the conversation between process operator and the computer covering this event is presented in Figure 6.7, where the italics are the operator's reply.

6.3.4 Performance of the self-learning fault diagnosis system

The self-learning fault diagnosis system has been tested for several threshold values in a similar way as in the above example, and the results of these experiments are shown in Table 6.2. In the first three cases, faults are detected but no diagnosis result is presented. Then the self-learning fault diagnosis system immediately reasons its behaviour. Any inappropriate thresholds are found and the diagnosis result under the new thresholds is presented. By this means, the fault diagnosis is not delayed by the inappropriate settings of certain threshold values. In the last three cases, the diagnosis results are wrong, as found by the process operator. After being informed that the diagnosis result is wrong, the self-learning fault diagnosis system examines its own behaviour, and finds any inappropriate threshold values and the diagnosis result under the new threshold values. From safety considerations, the self-learning fault diagnosis system will not make any changes in threshold but makes recommendations to operators, who can change the threshold values based on his own judgment.

6.4 Conclusions

This chapter describes a self-learning fault diagnosis system based on qualitative modelling. As the qualitative model based approach is gaining its popularity in process fault diagnosis, the technique presented in this chapter could have its practical values. The ability for reasoning its own behaviour is a desirable property for any future generation fault diagnosis system. With such a property, the fault diagnosis system will become more autonomous; in that it can explain its own behaviour, aid a developer with debugging, and adapt its behaviour to a changing environment. Through reasoning its own behaviour, the fault diagnosis system can improve its own performance over time and, hence, exhibits self-learning attributes.

By recording the problem solving history, all the intermediate problem solving states are available. Since the model of a diagnosis system is also available, learning can be achieved by reasoning the behaviour of a fault diagnosis system from its model. This fundamental idea may also be applied in other knowledge-based problem solving systems.

Apart from inappropriate parameters, there are other reasons for failures in diagnosis, such as incorrect models and incorrect generating of hypothesis, which

Table 6.2: Performance of the self-learning fault diagnosis system

inappropriate threshold values	initiated fault	failure in diagnosis	result of self-learning
$CT1(12) = 0.3cm$	level sensor 2 fail	fault is detected but not diagnosed	$CT2(12) = 0.14cm$ level sensor 2 fail
$CT1(1) = 0.3°C$	hot water control valve fail	same as above	$CT2(1) = 0.12°C$ hot water control valve fail
$CT1(3) = 0.2cm$	cold water control valve fail	same as above	$CT2(3) = 0.17cm$ cold water control valve fail
$CT1(16) = 6.0°C$	temp. sensor 1 fail	wrong diagnosis: hot water control valve fail	$CT2(16) = 5.0°C$ temp. sensor 1 fail
$CT1(17) = 2.0cm$	level sensor 1 fail	wrong diagnosis: cold water control valve fail	$CT2(17) = 3.0cm$ level sensor 1 fail
$CT1(9) = 4.0°C$ $CT1(14) = 0.6°C$	temp. sensor 2 fail	wrong diagnosis: hand valve 1 blocked	$CT(14) = 0.43°C$ temp. sensor 2 fail

are not concerned here. These could be investigated in future researches.

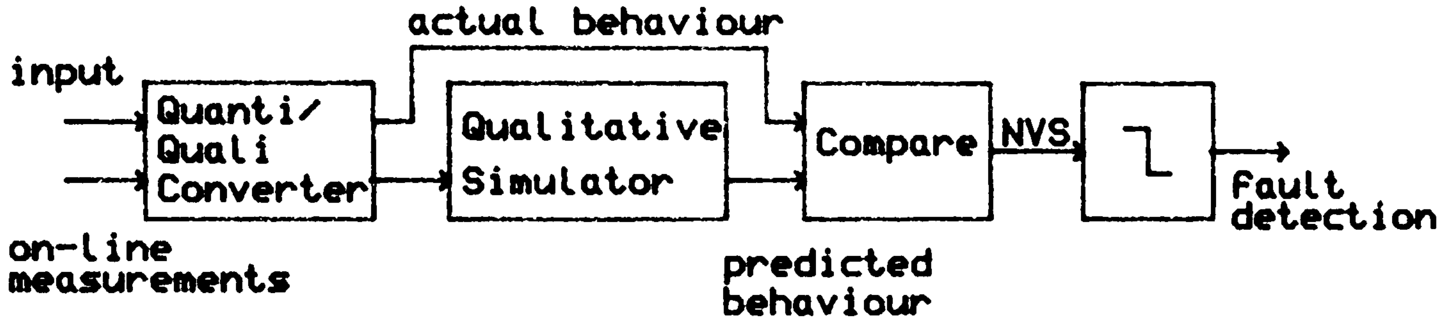


Fig.6.1 Fault detection

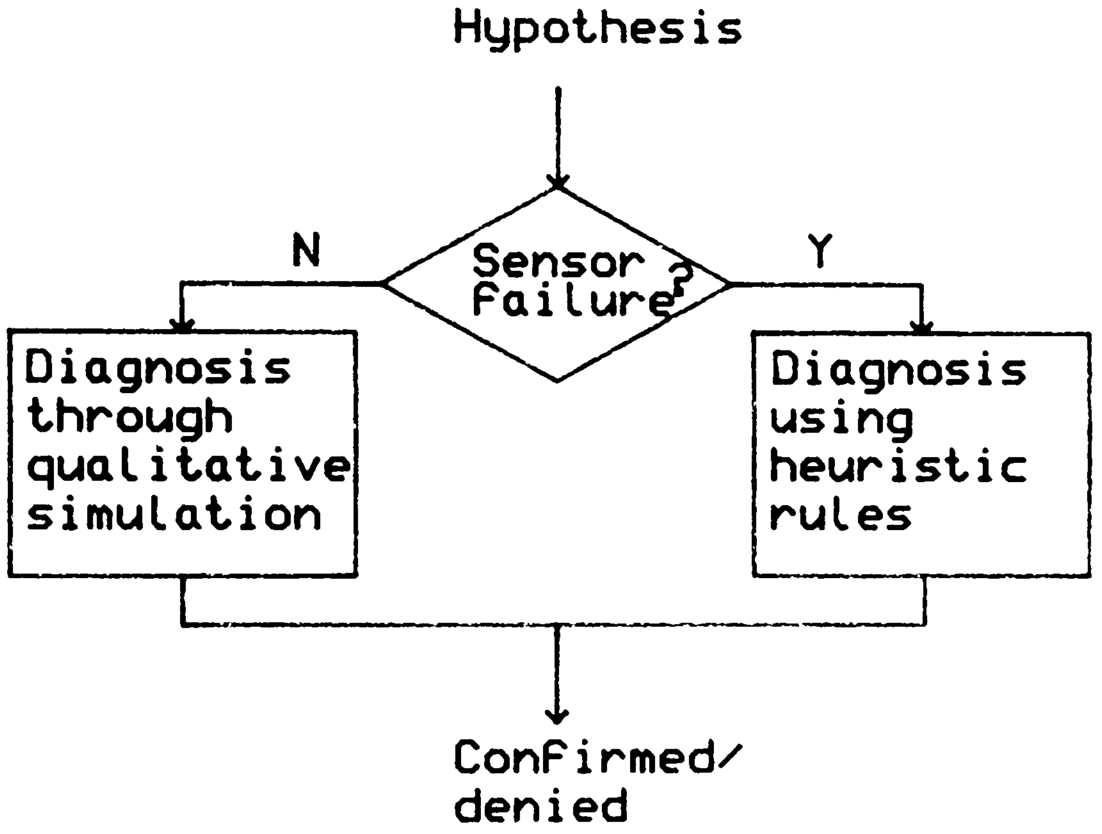

Hypothesis
N
Sensor Failure?
Y
Diagnosis through qualitative simulation
Diagnosis using heuristic rules
Confirmed/ denied

Fig.6.2 Diagnosis of different types of failures

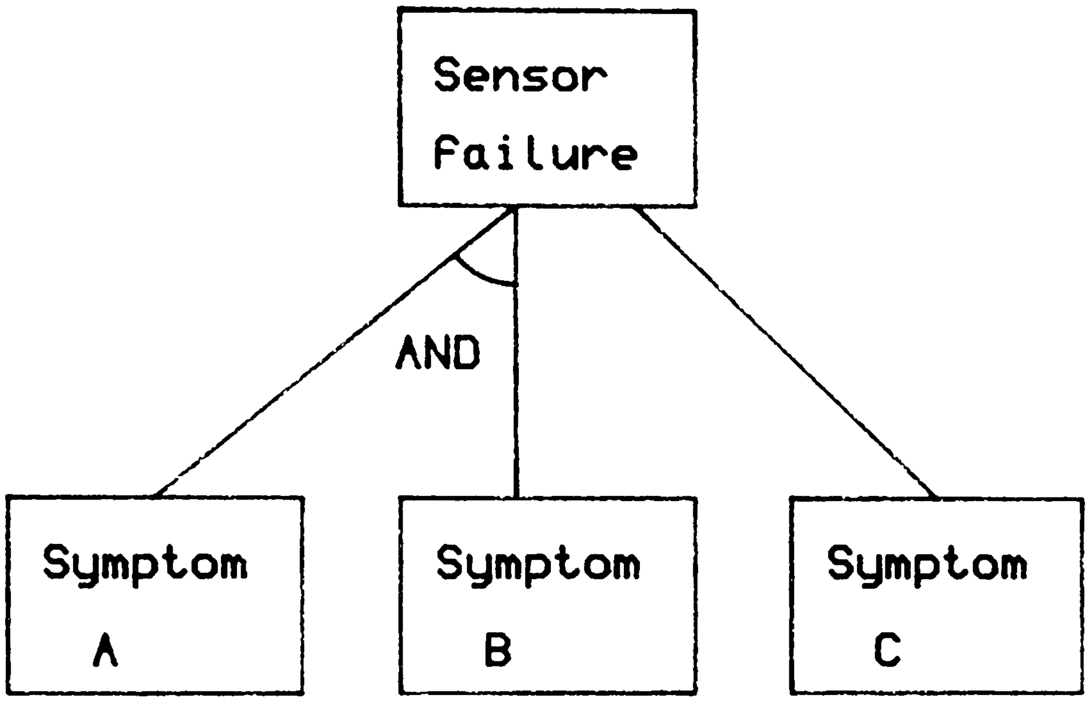


Fig.6.3 Diagnosis of sensor failures

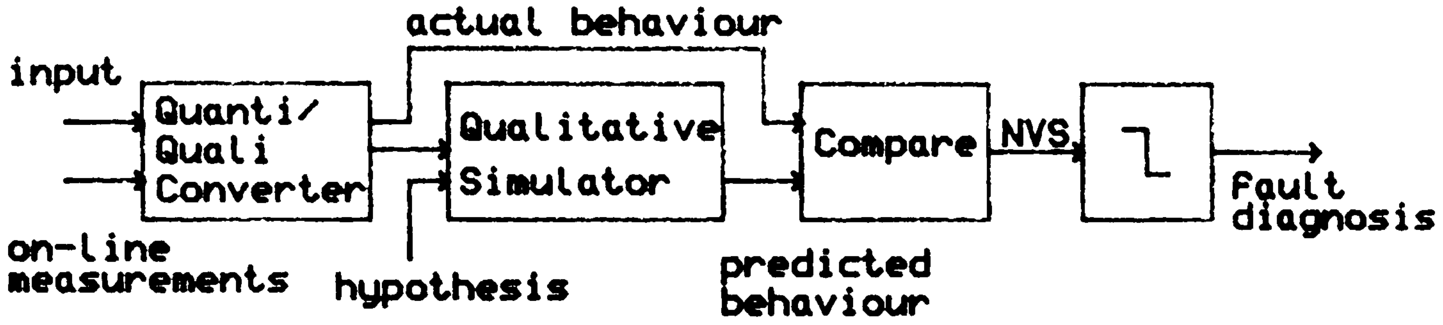
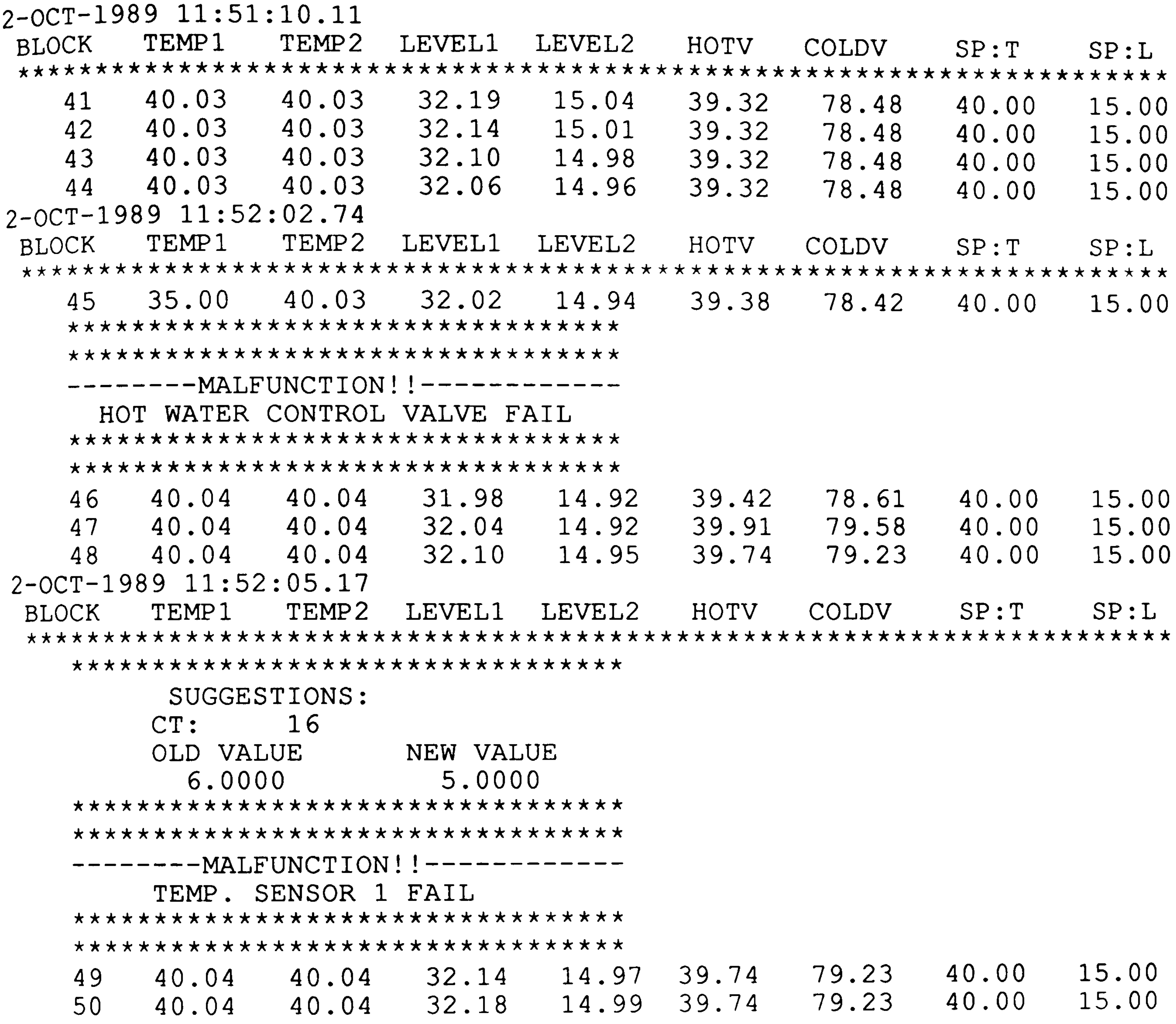


Fig.6.4 Fault diagnosis

```
TYPE:  "1" TO CHANGE SET POINTS
       "2" TO LIST/CHANGE OTHER PARAMETERS
       "3" TO CHANGE PRINT MODE
       "4" TO CHANGE BBC MONITORING MODE
       "5" TO SHUT DOWN
       "6" TO EXIT CONVERSATIONAL MODE
       "7" TO SET REPAIR FLAG
       "8" TO CHANGE THRESHOLD VALUES
   8
     CHANGE CURRENT THRESHOLDS?
y
ENTER PARAMETER NUMBER>
16
ENTER NEW VALUE>
6.0
CHANGE CURRENT THRESHOLDS?
n
CHANGE HIGH LIMIT OF THRESHOLDS?
n
CHANGE LOW LIMIT OF THRESHOLDS?
n

TYPE:  "1" TO CHANGE SET POINTS
       "2" TO LIST/CHANGE OTHER PARAMETERS
       "3" TO CHANGE PRINT MODE
       "4" TO CHANGE BBC MONITORING MODE
       "5" TO SHUT DOWN
       "6" TO EXIT CONVERSATIONAL MODE
       "7" TO SET REPAIR FLAG
       "8" TO CHANGE THRESHOLD VALUES
     6
```

Figure 6.5 Changing a threshold value



```
2-OCT-1989 11:51:10.11
 BLOCK   TEMP1   TEMP2  LEVEL1  LEVEL2   HOTV   COLDV    SP:T    SP:L
*********************************************************************
    41   40.03   40.03   32.19   15.04   39.32   78.48   40.00   15.00
    42   40.03   40.03   32.14   15.01   39.32   78.48   40.00   15.00
    43   40.03   40.03   32.10   14.98   39.32   78.48   40.00   15.00
    44   40.03   40.03   32.06   14.96   39.32   78.48   40.00   15.00
2-OCT-1989 11:52:02.74
 BLOCK   TEMP1   TEMP2  LEVEL1  LEVEL2   HOTV   COLDV    SP:T    SP:L
*********************************************************************
    45   35.00   40.03   32.02   14.94   39.38   78.42   40.00   15.00
    *********************************
    *********************************
    --------MALFUNCTION!!------------
      HOT WATER CONTROL VALVE FAIL
    *********************************
    *********************************
    46   40.04   40.04   31.98   14.92   39.42   78.61   40.00   15.00
    47   40.04   40.04   32.04   14.92   39.91   79.58   40.00   15.00
    48   40.04   40.04   32.10   14.95   39.74   79.23   40.00   15.00
2-OCT-1989 11:52:05.17
 BLOCK   TEMP1   TEMP2  LEVEL1  LEVEL2   HOTV   COLDV    SP:T    SP:L
*********************************************************************
    *********************************
          SUGGESTIONS:
         CT:     16
         OLD VALUE      NEW VALUE
           6.0000         5.0000
    *********************************
    *********************************
    --------MALFUNCTION!!------------
         TEMP. SENSOR 1 FAIL
    *********************************
    *********************************
    49   40.04   40.04   32.14   14.97  39.74   79.23   40.00   15.00
    50   40.04   40.04   32.18   14.99  39.74   79.23   40.00   15.00
```

Figure 6.6 On-line displayed information

```
TYPE:   "1" TO CHANGE SET POINTS
        "2" TO LIST/CHANGE OTHER PARAMETERS
        "3" TO CHANGE PRINT MODE
        "4" TO CHANGE BBC MONITORING MODE
        "5" TO SHUT DOWN
        "6" TO EXIT CONVERSATIONAL MODE
        "7" TO SET REPAIR FLAG
        "8" TO CHANGE THRESHOLD VALUES
7
TYPE "0" FOR FINISHING REPAIR
     "1" FOR UNDER REPAIRING
     "2" FOR INCORRECT DIAGNOSIS
2

TYPE:   "1" TO CHANGE SET POINTS
        "2" TO LIST/CHANGE OTHER PARAMETERS
        "3" TO CHANGE PRINT MODE
        "4" TO CHANGE BBC MONITORING MODE
        "5" TO SHUT DOWN
        "6" TO EXIT CONVERSATIONAL MODE
        "7" TO SET REPAIR FLAG
        "8" TO CHANGE THRESHOLD VALUES
6
```

Figure 6.7 Inform a wrong diagnosis

Chapter 7

Fault diagnosis by the combined use of deep knowledge and heuristics — with the heuristics learned from deep knowledge based diagnosis

7.1 Introduction

As mentioned in Chapter 2, expert systems for industrial process fault diagnosis can generally be divided into two categories: a shallow knowledge based approach and a deep knowledge based approach. In the first category the knowledge base contains heuristic rules which encode the experiences of process operators. This type of expert system can usually diagnose faults very efficiently because heuristics can provide valuable short cuts (Lapointe *et al* 1989, Moor and Kramer 1986). Lapointe *et al* (1989) developed an expert system for waste water treatment process diagnosis — BIOEXPERT, in which shallow knowledge is used for diagnosing the more common faults. Since the knowledge base does not contain any deep knowledge, such as the knowledge about system structure and component functions, it may have difficulties when dealing with novel faults and infrequently occurred faults. In contrast, in the deep knowledge based approach, the knowledge base contains information on system structures and unit functions as well as physical laws governing the process. With such a knowledge base, fault diagnosis can be carried out with greater reliability.

However, the diagnostic efficiency is affected by its detailed knowledge base, because the diagnosis system needs to explore the entire causal path from a failed component to the observed abnormalities.

To enhance both efficiency and reliability, a combination of the two approaches should be considered. There is a trend towards building fault diagnosis systems using both shallow and deep knowledge (Lapointe *et al* 1989, Moor and Kramer 1986, Venkatasubramanian and Rich 1988). Venkatasubramanian and Rich (1988) discuss a fault diagnosis system for a chemical process using both types of knowledge. They propose a two-tier architecture for integrating compiled and deep level knowledge in that the process specific compiled knowledge is stored at the top tier, while the lower tier holds deep knowledge. During diagnosis, the compiled knowledge is invoked first. If a diagnosis result cannot be obtained from the compiled knowledge, the diagnosis will drop down to the deep level knowledge.

To reduce the effort of encoding and debugging diagnostic heuristics from diagnostic experts, machine learning techniques (Michalski *et al* 1983, Forsyth and Rada 1986) can be used to automatically acquire diagnostic heuristics. Recently several researchers have attempted to incorporate a learning mechanism into process fault diagnosis systems to make them more intelligent (Ishida 1988, Pazzani 1986, 1987, Rich and Venkatasubramanian 1989). In Pazzani's approach (Pazzani 1986, 1987), a set of initially developed heuristic rules are used to propose a hypothesis when an abnormal condition is encountered, and a deep model is then used to confirm this hypothesis. If it cannot be confirmed, then the heuristic rule which proposed this hypothesis is considered to have failed and it is revised by adding additional terms to its condition part to limit its applicability. This is called failure-driven learning since learning is initiaed when a hypothesis failure occurs. Through this failure-driven learning, the existing heuristic rules can be refined but there may exist situations where there are no heuristic rules corresponding to some failures, especially failures which occur infrequently. In such situations, it would be desirable that the system can still diagnose the fault and learn a new heuristic rule. This is not addressed in Pazzani's approach (Pazzani 1986, 1987). Rich and Venkatasubramanian (1989) discuss a causality-based failure-driven learning approach. In their approach, when a heuristic rule fails to propose the right hypothesis, the rule is revised and the system will drop down to deep knowledge based diagnosis, and it could learn a new heuristic rule. This method is developed for off-line diagnosis as can be seen from the context of Venkatasubramanian and Rich (1988), and Rich and Venkatasubramanian (1989). The condition parts of some heuristic rules include the negation of the failures of some other compo-nents, and this information is obtained from the operator. The aim of these failure-driven learning approaches is mainly to refine the existing heuristic rules.

Apart from these failure-driven learning approaches, Ishida (1988) demonstrates that diagnostic heuristics can be learnt from qualitative simulation of the process behaviour. Simulation of a process is conducted by inserting a fault as a disturbance to the qualitative model. The qualitative deviations of certain process variables are calculated and compiled to form a rule corresponding to this fault.

In this research, an on-line fault diagnosis system which uses both deep knowledge and heuristics is investigated. During diagnosis, the system will first invoke the heuristic rules to propose a hypothesis. If a hypothesis can be proposed, then a deep model is used to discriminate this hypothesis. Otherwise, the diagnosis is based entirely on the deep model. The fault diagnosis system will test a set of candidate faults by inserting each fault as a disturbance to the qualitative model. The candidate which can explain the observed abnormalities is taken as the diagnosis result. The system can learn new diagnostic heuristic rules and refine existing ones during diagnosis. Learning is initiated not only when a heuristic rule proposes a wrong hypothesis, but also when there is not a heuristic rule corresponding to a successful diagnosis. Initially, there can be a few, or even no, heuristic rules and, during diagnosis, the system will continuously learn heuristics such that rules can be gradually built up.

In the next section, diagnosis using both heuristics and deep knowledge is described. Section 7.3 describes the procedure of learning new diagnostic rules and refining existing ones. An illustrative application to the on-line fault diagnosis of the mixing process is presented in Section 7.4. Section 7.5 describes the application to the CSTR system. The last section contains some concluding remarks.

7.2 Fault diagnosis using both heuristics and deep knowledge

Taking account of issues of efficiency and reliability, an on-line system which uses both heuristics and deep knowledge to diagnose faults has been investigated. The heuristics, in the form of rules, are used to propose a hypothesis. The deep knowledge, in the form of a deep qualitative model, is used to confirm the proposed hypothesis. Therefore, when abnormalities occur in the measurements, the diagnosis system will match the observed abnormalities with the condition parts of heuristic rules, and the rule whose condition part matches the observed abnormalities is used to propose a hypothesis. The qualitative model is then used to predict the behaviour of the process under this hypothesis by means of qualitative simulation techniques (Bobrow 1984). This prediction is compared with the actual behaviour of the process and, depending on whether they agree or not, the hypothesis is confirmed or denied. This is referred to as the "hypothesis-test strategy" (Moor and Kramer 1986). By this means, diagnostic efficiency is achieved by the use of heuristic rules and diagnostic reliability is ensured by the use of a deep qualitative model.

The heuristic rules may be incomplete and some of them may be incorrect such that the hypothesis proposed may later be denied by the deep model of the process, or no hypothesis can be generated by the heuristic rules. This is referred to as "failures in using heuristic rules". When such cases are encountered, the fault diagnosis system will rely on the deep model based approach. It will use a hypothesis-test strategy to test a set of candidate failures. The desired behaviour of the process corresponding to each candidate failure is predicted through qualitative simulation and is compared with the actual behaviour of the process, and the candidate which can explain the observed abnormalities is taken as the diagnosis result. Therefore, the incompleteness in heuristic rules will not obstruct the diagnosis.

Since the incompleteness in the rules will reduce the diagnostic efficiency, it would be desirable that the fault diagnosis system can learn heuristic rules itself. This is also desirable from the point of view of easing the task of knowledge acquisition, which often needs considerable effort because process experts usually have little knowledge about knowledge engineering. Furthermore, it is also often difficult for a knowledge engineer to fully understand the operation of a specific process, and this issue is often referred to as the "knowledge engineering bottleneck" (Moor and Kramer 1986, Price and Lee 1988). By means of machine learning techniques, the diagnosis system can automatically build up its heuristic rule base and, hence, the diagnostic efficiency will be gradually improved.

The fault diagnosis system described in this chapter is designed to fulfill the above requirement. Initially the heuristic rule base contains a limited number of heuristic rules, or may even be empty. After each diagnosis, in which the diagnosed fault is not proposed by the heuristic rules, the system will learn a new rule by recognising any significant patterns in the deviations of measurements and compiling them to form a heuristic rule. By such means, the heuristic rule base will gradually be assembled.

During diagnosis, the number of occurrences of each fault is recorded, and the heuristic rules corresponding to the frequently occurred failures are arranged at the top of the rule base. Therefore, the fault diagnosis system can diagnose more frequently occurred faults more efficiently.

7.3 Learning diagnostic heuristic rules

After a successful diagnosis, the system will examine the result and decide if learning should be initiated, which occurs when failures arise in using heuristic rules. These failures include the following situations: 1), there is no heuristic rule corresponding to the diagnosed fault and no heuristic rule is employed; 2), there is no heuristic rule corresponding to the diagnosed fault but one of the heuristic rules is mistakenly used; 3), there is a heuristic rule corresponding to the diagnosed fault, but none of these rules is used; 4), there is a heuristic rule corresponding to the diagnosed fault, but it is not employed and, instead, one of the other rules is erroneously employed.

For the first two situations, a new heuristic rule needs to be learnt from the successful diagnosis since there is no rule corresponding to the diagnosed fault and, furthermore, for the second situation, apart from learning this new heuristic rule, the rule which proposed a wrong hypothesis should also be refined so that it will not erroneously be employed in future similar situations. For the last two situations, the condition part of the existing heuristic rule corresponding to the diagnosed fault should be revised such that it can match the current condition and, hence, propose the correct hypothesis in future similar conditions. For the last situation, in addition, the incorrectly employed rule should be revised such that its applicability should be limited. To summarise, there are two basic learning tasks, namely to learn new diagnostic heuristic rules and to refine existing rules.

7.3.1 Learning new diagnostic heuristic rules

After a successful diagnosis a new rule can be constructed from the recorded on-line measurements and controller outputs used for this diagnosis and its result. The consequence part of the rule is simply the diagnosed fault while the condition part of this rule contains symptoms associated with this fault. These symptoms are the qualitative increments (increase, steady, and decrease) of certain measurements and controller outputs over an interval.

The diagnostic heuristic rules are organised in the form of a table, as shown in Figure 7.1, where each row corresponds to a specific rule and the empty rows are used to hold new rules. Hanakuma (1989) describes a fault diagnosis system for use at petrochemical plants, where diagnostic rules are stored in the form of a table similar to Figure 7.1, and the table is called CSM (Cause-Symptom Matrix). The last box in each row contains the consequence part of the rule. The other boxes correspond to the on-line measurements and controller outputs used in diagnosis and are used for holding symptoms which comprise the conjunctive qualitative increments of the on-line measurements and controller outputs. Each of these boxes is filled by one of the signs, +, 0, −, and *, to represent increase, steady, decrease, and unused, respectively. Since some measurements are not needed in the diagnosis of a specific fault, the boxes corresponding to these measurements are filled by "*" to indicate that they are not needed. For example, the first row of the table in Figure 7.1 can be interpreted as:

IF H_1 increases
H_2 increases
Q_c decreases
Q_h increases
THEN Cold water control valve fails

Learning a heuristic rule is simply performed by filling the empty boxes in the the first empty row. This is essentially the signature table method (Forsyth and Rada 1986) in machine learning. The box for holding the conclusion part of the heuristic rule is simply filled with the diagnosed fault. The symbols in the other boxes are entered by comparing the increments of the on-line measurements and controller outputs over an interval with the correspondingly pre-defined thresholds. By this means the qualitative increments of all the measurements and controller outputs can be obtained. Some of the measurements may not be affected by this fault and, therefore, the boxes corresponding to these measurements should be filled with "*". The problem here is how can the computer know which measurements are not affected by a specific fault. Since the hypothesis is discriminated by comparing the predicted behaviour of the process, obtained from qualitative simulation, with its actual behaviour, these predictions can be used to guide the determination of symptoms. If some variables' predicted qualitative increments under a fault are identical with the predictions under normal operating conditions, then these variables are assumed not to be affected by this fault, and the boxes corresponding to these measurements should be filled in with "*". This information can also be input by the process operators.

This method is similar to Ishida's method for learning diagnostic rules from qualitative simulation (Ishida 1988). In his approach, since the deviations of certain variables caused by a fault are obtained from qualitative simulation, ambiguity needs to be given crucial consideration. In the method presented here, since all these values are directly calculated from on-line measurements, there is no ambiguity, but they may be affected by measurement noise and disturbances. Therefore, it is helpful to take several sets of measurements when determining the qualitative deviations.

7.3.2 Refining the existing heuristic rules

There are two situations in which some of the existing heuristic rules should be refined. One is that the hypothesis proposed by a heuristic rule is denied by the deep model, and another is that the diagnosed fault is not proposed by the existing corresponding heuristic rule. For the first situation, the reason may be that the heuristic rule which proposed a wrong hypothesis is too general in that its condition part does not contain sufficient symptoms of the associated fault. Therefore, the value of the boxes marked with "*" should be redetermined to see if some of these values can be changed such that this rule will not be employed in future similar cases. The failure driven-learning described in (Pazzani 1986, 1987) and (Rich and Venkatasubramanian 1989) are mainly concerned with this type of failure. In (Pazzani 1986, 1987) and (Rich and Venkatasubramanian 1989), the initially developed diagnostic heuristic rules are crude in that only some of the symptoms are included in the condition parts of the diagnostic rule and, therefore, may often generate a wrong hypothesis.

One reason for the second situation is that some symptoms in the condition part of the failed heuristic rule may be incorrect due to inappropriate threshold values being used for determining the symptoms and, therefore, they should be corrected. This can be performed by changing the related thresholds in a certain range such that the condition part of this rule can match the current situation. Another possible reason is that the fault may behave in different ways. For example, a control valve may fail high in that its output flow is higher than the value corresponding to its input, or fail low in that its output flow is lower than the value corresponding to its input. For this type of failure, a new rule is initially learnt for this immediate diagnosis as described previously, and then its condition part is compared with that of the previous rule corresponding to the same failure. Then if only few terms are different they can be regarded as unused terms and, therefore, the corresponding boxes are filled with "*", and the new and old rules are merged together. If there are many different terms between the condition parts of the new and old rules, then it is considered that the fault behaves in different ways and the two rules represent two different forms of the same fault. In this case, both of the rules are retained.

If the heuristic rules are presented by the designer, as in (Pazzani 1986, 1987) and (Rich and Venkatasubramanian 1989), then the condition parts of the rules generally contain less symptoms and, in this case, the first type of failure may often occur. However, for rules learned on-line, the condition parts often include many terms and, in this case, the second type of failure may often occur.

7.3.3 Rearrangement of heuristic rules

After learning a new rule or refining an old one, the heuristic rules are rearranged in the order of the frequencies that the related faults occurred, such that the rule related to the most frequently occurred fault will be on the top of the rule base. Through such a dynamically reordering of the heuristic rules, the system will be efficient in diagnosis of frequently occurred faults. This is in contrast to the method of Rich and Venkatasubramanian (1989), which assumes that the initially developed diagnostic heuristic rules are related with frequently occurred faults and, therefore, in their method there is no reordering of the diagnostic rules.

7.4 Application to the on-line fault diagnosis of the mixing process

7.4.1 Fault diagnosis of a mixing process based on its qualitative model

The on-line learning method described above has been incorporated into the fault diagnosis system for the mixing process described in Chapter 5. The modified diagnosis system will use both heuristic rules, which can be learnt on-line, and a deep qualitative model. When abnormal behaviour is detected, the diagnosis system will first try to generate a hypothesis using heuristic rules and then to confirm or reject the hypothesis through qualitative simulation. If no hypothesis can be generated, then the fault diagnosis system will work in the same way as that presented in Chapter 5. During diagnosis, the system can continuously learn new diagnostic rules and refine existing rules.

7.4.2 Learning heuristic rules from deep knowledge based diagnosis

Unlike failures of other components, sensor failures will not usually result in a relatively fixed symptom and, therefore, currently, the diagnosis system is designed for learning heuristic rules related only to the failures of non-sensor components. In the mixing process these failures would be: the blockage of hand valve 1 and hand valve 2, and the failure of hot and cold water control valves. Initially, the heuristic rule base is empty and, after a successful diagnosis of one of these faults, a corresponding diagnostic heuristic rule is learnt. Once such a rule exists, the system will try to use this rule to generate a hypothesis in the next diagnosis. The heuristic rule base will thus be gradually assembled.

7.4.3 Case studies

After the mixing process has been operating at steady state, a failure, in which the cold water control valve fails and gives a high output flow, is initiated. The on-line measurements and the controller outputs covering this event are shown in Figure 7.2, where Figure 7.2 (a) illustrates the data of level measurements, Figure 7.2 (b) shows the temperature measurements, and the controller outputs are presented in Figure 7.2 (c). Before 480 seconds, the process is operating at steady state. Then the failure is initiated, and it is observed that after the failure has been diagnosed it is removed and the measurements return to their steady state values.

The diagnosis result is shown in Figure 7.3. Since initially there is no heuristic rule, the system generated two hypotheses to obtain the diagnosis result. After diagnosis, it learnt the rule:

IF
Level in tank 1 increases
Level in tank 2 increases
Cold water flow decreases
Hot water flow increases
THEN
Cold water control valve fails

After the process has returned to its steady state, the same failure is initiated again and the resulting on-line measurements and the controller outputs are shown in Figure 7.4. The diagnosis result is shown in Figure 7.3, and it can be seen that this time the diagnosis result is directly proposed by the on-line learnt heuristic rule. After diagnosis, the failure is corrected and all the measurements return to their steady state values.

After the process settles down, another fault in which hand valve 2 becomes blocked is initiated. The resulting on-line measurements and controller outputs are shown in Figure 7.5 and the diagnosis result is given in Figure 7.6. It can be seen that six hypotheses have been generated to diagnose this fault and, after diagnosis, the system learnt the new rule:

IF
Level in tank 1 steady
Level in tank 2 increases
Cold water flow decreases
Hot water flow decreases
THEN
Hand valve 2 is blocked

After diagnosis, the failure is corrected and Figure 7.5 shows the measurements return to their steady state values. The same failure is initiated again after the process has settled down, and the on-line measurements and the controller outputs are shown in Figure 7.7, with the diagnosis result given in Figure 7.6. It can be seen that this time the diagnosed fault is directly proposed by the newly learnt rule.

7.5 Application to the fault diagnosis of the CSTR system

7.5.1 Learning diagnostic rules for the CSTR system

A similar modification is also made to the fault diagnosis system for the CSTR system described in Chapter 5. The modified system uses both heuristic rules and a qualitative model in a similar manner to that described in the previous section.

In the CSTR system, there are a lot of measurements and possible faults and, in this case, the learning method described in the previous section may not be efficiently applied since the large number of measurements could make the condition parts of the learnt rules very bulky. To overcome this, the CSTR system is decomposed into

Table 7.1: Performance of the diagnosis system without heuristic rules

Initiated faults	No. of hypotheses generated & tested
Pipe 1 is blocked	8
External feed flow rate high	9
Pipe 2 or 3 is blocked or pump fails	16
Pipe 10 or 11 is blocked or control valve 1 fails low	11
External feed reactant temperature is abnormal	12
Control valve 2 fails high	4
Pipe 7 or 8 or 9 is blocked, or control valve 2 fails low	5
Control valve 1 fails high	13
Pipe 4 or 5 or 6 is blocked, or control valve 3 fails low	6
Control valve 3 fails high	7
External feed reactant concentration too low	14

two subsystems as described in Chapter 5. The diagnosis system will learn locally valid diagnostic rules for each subsystem. The rules for a subsystem is only valid if the qualitative model of that subsystem is violated.

7.5.2 Performance of the diagnosis system

Simulation studies have been performed to investigate the performance of the diagnosis system. Initially, the heuristic rule base is empty and, in this case, the performance is the same as that described in Chapter 5. All the failures, excluding sensor failures, were initiated individually. Table 7.1 shows the numbers of hypotheses generated during diagnosis. It can be seen that a large number of hypotheses have to be generated and tested to diagnose some faults.

After these faults have been initiated and diagnosed, the diagnosis system learnt a diagnostic rule for each fault. These rules are arranged into two groups corresponding to the two subsystems of the CSTR system, and are listed below:

Rule set 1: (for the first subsystem)

Rule 1.1

IF

FLOW OF STREAM 1	DECREASE
TEMP. OF FLOW STREAM 1	STEADY
EXTERNAL FEED CONC.	STEADY
LEVEL IN REACTOR	DECREASE
TEMP. IN REACTOR	INCREASE
FLOW OF STREAM 4	DECREASE
CONC. CA AFTER REACTION	DECREASE
INPUT SIGNAL TO VALVE 1	INCREASE

THEN

PIPE 1 BLOCKED

Rule 1.2

IF

FLOW OF STREAM 1	INCREASE
TEMP. OF FLOW STREAM 1	STEADY
EXTERNAL FEED CONC.	STEADY
FLOW OF STREAM 4	INCREASE
CONC. CA AFTER REACTION	INCREASE
INPUT SIGNAL TO VALVE 1	DECREASE

THEN

EXTERNAL FEED FLOW TOO HIGH

Rule 1.3

IF

FLOW OF STREAM 1	STEADY
TEMP. OF FLOW STREAM 1	STEADY
EXTERNAL FEED CONC.	STEADY
LEVEL IN REACTOR	INCREASE
TEMP. IN REACTOR	DECREASE
FLOW OF STREAM 4	DECREASE
CONC. CA AFTER REACTION	STEADY
INPUT SIGNAL TO VALVE 1	DECREASE

THEN

PIPE 2 OR 3 BLOCKED, OR PUMP FAIL

Rule 1.4

IF

FLOW OF STREAM 1	STEADY
TEMP. OF FLOW STREAM 1	STEADY
EXTERNAL FEED CONC.	STEADY
LEVEL IN REACTOR	INCREASE
TEMP. IN REACTOR	DECREASE
FLOW OF STREAM 4	DECREASE
CONC. CA AFTER REACTION	INCREASE
INPUT SIGNAL TO VALVE 1	DECREASE

THEN

PIPE 10 OR 11 BLOCKED, OR CONTROL VALVE 1 FAIL LOW

Rule 1.5

IF

FLOW OF STREAM 1	STEADY
TEMP. OF FLOW STREAM 1	INCREASE
EXTERNAL FEED CONC.	STEADY
LEVEL IN REACTOR	STEADY
TEMP. IN REACTOR	INCREASE
FLOW OF STREAM 4	STEADY
INPUT SIGNAL TO VALVE 1	STEADY

THEN

EXTERNAL FEED TEMP. ABNORMAL

Rule 1.6

IF

FLOW OF STREAM 1	STEADY
TEMP. OF FLOW STREAM 1	DECREASE
EXTERNAL FEED CONC.	STEADY
LEVEL IN REACTOR	STEADY
TEMP. IN REACTOR	DECREASE
FLOW OF STREAM 4	STEADY

```
    INPUT SIGNAL TO VALVE 1          STEADY
    THEN
    EXTERNAL FEED TEMP. ABNORMAL

Rule 1.7
    IF
    FLOW OF STREAM 1                 STEADY
    TEMP. OF FLOW STREAM 1           STEADY
    EXTERNAL FEED CONC.              STEADY
    FLOW OF STREAM 4                 INCREASE
    CONC. CA AFTER REACTION          STEADY
    INPUT SIGNAL TO VALVE 1          INCREASE
    THEN
    CONTROL VALVE 1 FAIL HIGH

Rule 1.8
    IF
    FLOW OF STREAM 1                 STEADY
    TEMP. OF FLOW STREAM 1           STEADY
    EXTERNAL FEED CONC.              DECREASE
    LEVEL IN REACTOR                 STEADY
    TEMP. IN REACTOR                 DECREASE
    FLOW OF STREAM 4                 STEADY
    CONC. CA AFTER REACTION          DECREASE
    INPUT SIGNAL TO VALVE 1          STEADY
    THEN
    EXTERNAL FEED CONC. TOO LOW
```

Rule set 2: (for the second subsystem)

```
Rule 2.1
    IF
    FLOW OF STREAM 2                 STEADY
    FLOW OF STREAM 5                 INCREASE
    PRESSURE OF FEED COLD WATER      STEADY
    TEMP. OF FEED COLD WATER         STEADY
    INPUT SIGNAL TO VALVE 3          STEADY
    INPUT SIGNAL TO VALVE 5          DECREASE
    THEN
    CONTROL VALVE 2 FAIL HIGH

Rule 2.2
    IF
    FLOW OF STREAM 2                 STEADY
    FLOW OF STREAM 5                 DECREASE
    PRESSURE OF FEED COLD WATER      STEADY
    TEMP. OF FEED COLD WATER         STEADY
    INPUT SIGNAL TO VALVE 3          STEADY
    INPUT SIGNAL TO VALVE 5          INCREASE
    THEN
    PIPE 7 OR 8 OR 9 BLOCKED, OR CONTROL
    VALVE 2 FAIL LOW

Rule 2.3
    IF
    FLOW OF STREAM 2                 DECREASE
    FLOW OF STREAM 5                 INCREASE
    PRESSURE OF FEED COLD WATER      STEADY
    TEMP. OF FEED COLD WATER         STEADY
    INPUT SIGNAL TO VALVE 3          INCREASE
    THEN
    PIPE 4 OR 5 OR 6 BLOCKED, OR CONTROL
    VALVE 3 FAIL LOW

Rule 2.4
    IF
    FLOW OF STREAM 2                 INCREASE
    FLOW OF STREAM 5                 STEADY
    PRESSURE OF FEED COLD WATER      STEADY
    TEMP. OF FEED COLD WATER         STEADY
    INPUT SIGNAL TO VALVE 3          DECREASE
    INPUT SIGNAL TO VALVE 5          STEADY
    THEN
```

CONTROL VALVE 3 FAIL HIGH

After these rules have been learnt, all those faults were initiated individually again and, in this case, they were all directly proposed by the learnt heuristic rules. This suggests an improvement in performance.

7.6 Conclusions

Diagnosis using both deep knowledge and heuristic rules would be a desirable way to enhance diagnostic efficiency and reliability. Valuable shortcuts for diagnosis may be available in the form of heuristic rules. A method for learning heuristic rules from deep knowledge based diagnosis has been presented in this chapter. This may be suitable for developing an on-line fault diagnosis system for a new process where heuristic rules for diagnosis may not be available or for a complex process where the rules cannot easily be obtained. For such applications, a deep knowledge based diagnosis system is first developed and, after each diagnosis, the significant patterns in the on-line measurements are recognised and are compiled to form a heuristic rule. By this means the heuristic rule base can be automatically assembled.

H_1	H_2	T_1	T_2	Q_c	Q_h	FAULT
+	+	*	*	-	+	Cold water valve fail
0	+	*	*	-	-	Hand valve 2 blocked

Figure 7.1 Heuristic rules

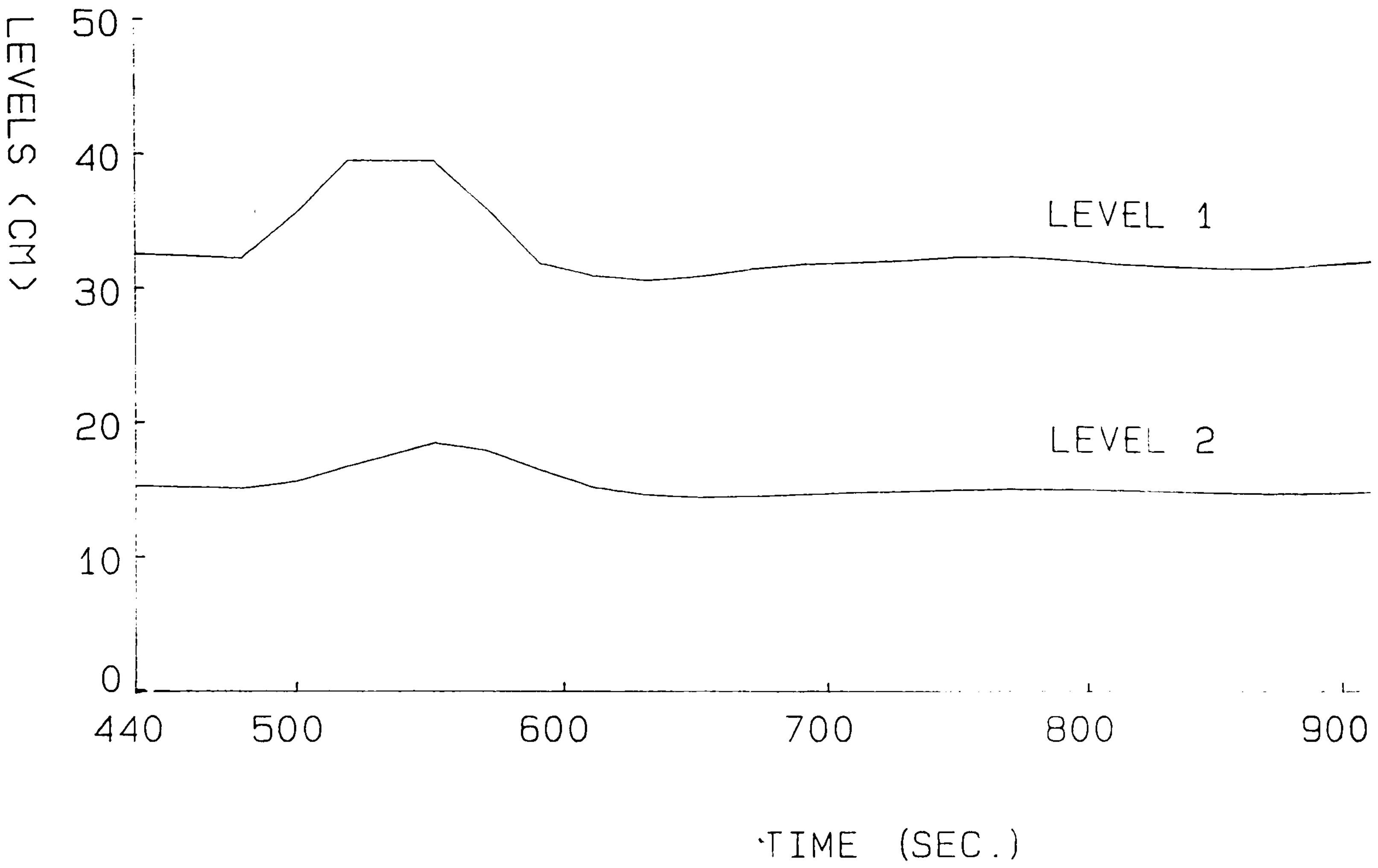


Figure 7.2(a) Level measurements for case study No.1 (Before learning)

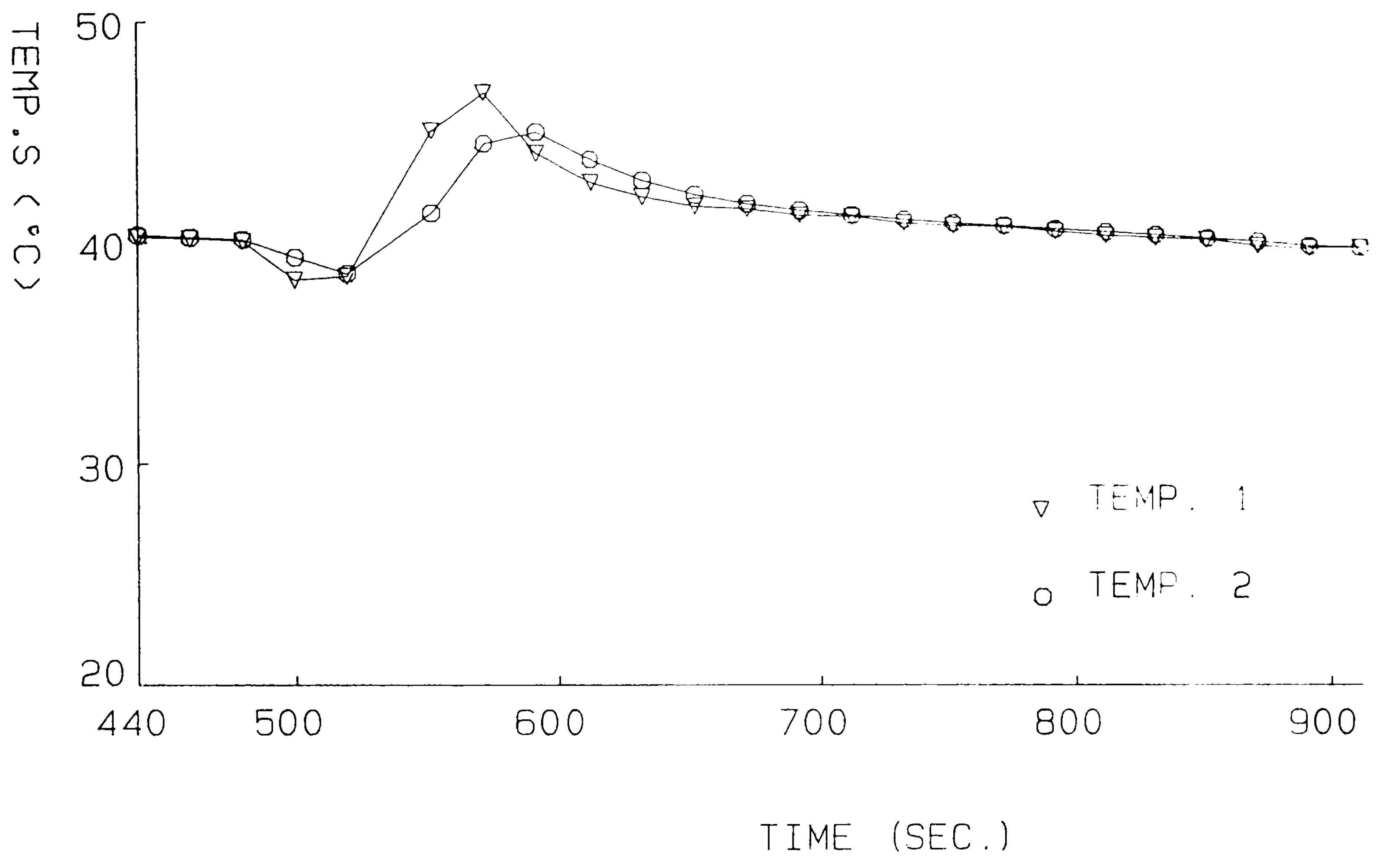


Figure 7.2(b) Temperature measurements for case study No.1 (Before learning)

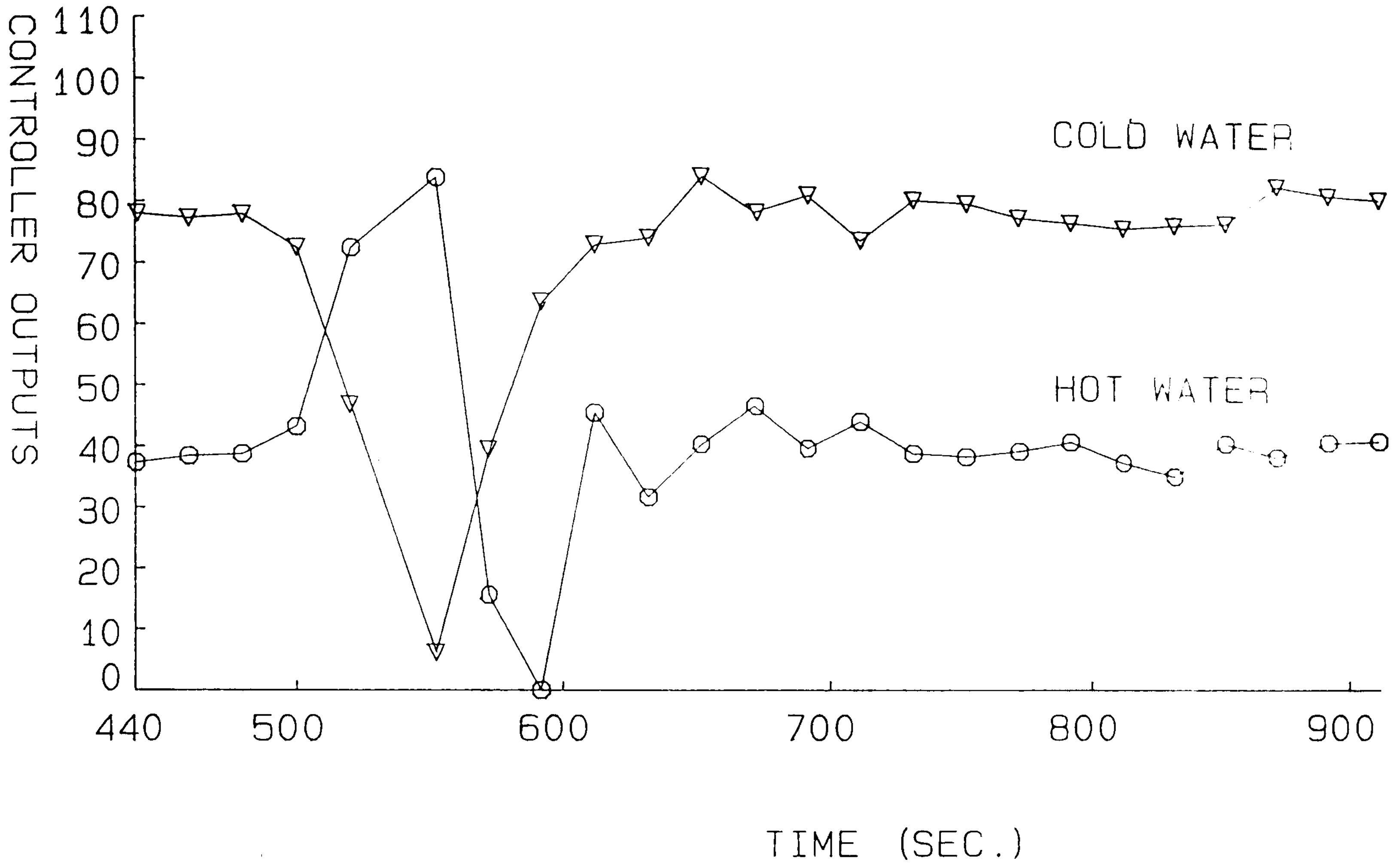


Figure 7.2(c) Controller outputs for case study No.1

(Before learning)

Performance before learning	Performance after learning
Initiated fault: Cold water control valve fail Proposed by a heuristic rule? No No. of hypothesis generated & tested: 2 On-line learnt rule: IF Level in tank 1 increase Level in tank 2 increase Cold water flow decrease Hot water flow increase THEN Cold water control valve fail	Initiated fault: Cold water control valve fail Proposed by a heuristic rule? Yes

Figure 7.3 Performance of the diagnosis system for case study No. 1

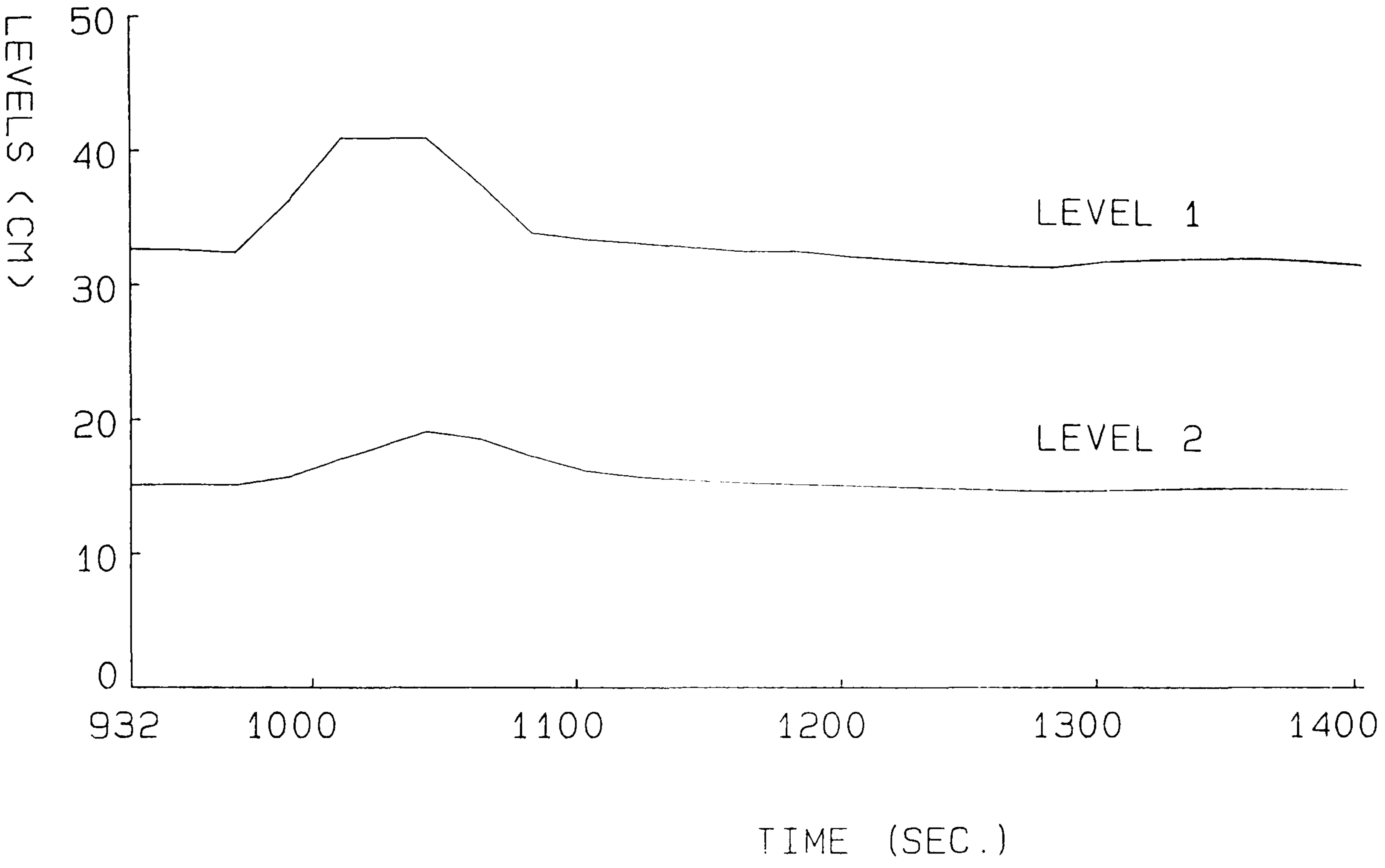


Figure 7.4(a) Level measurements for case study No.1

(After learning)

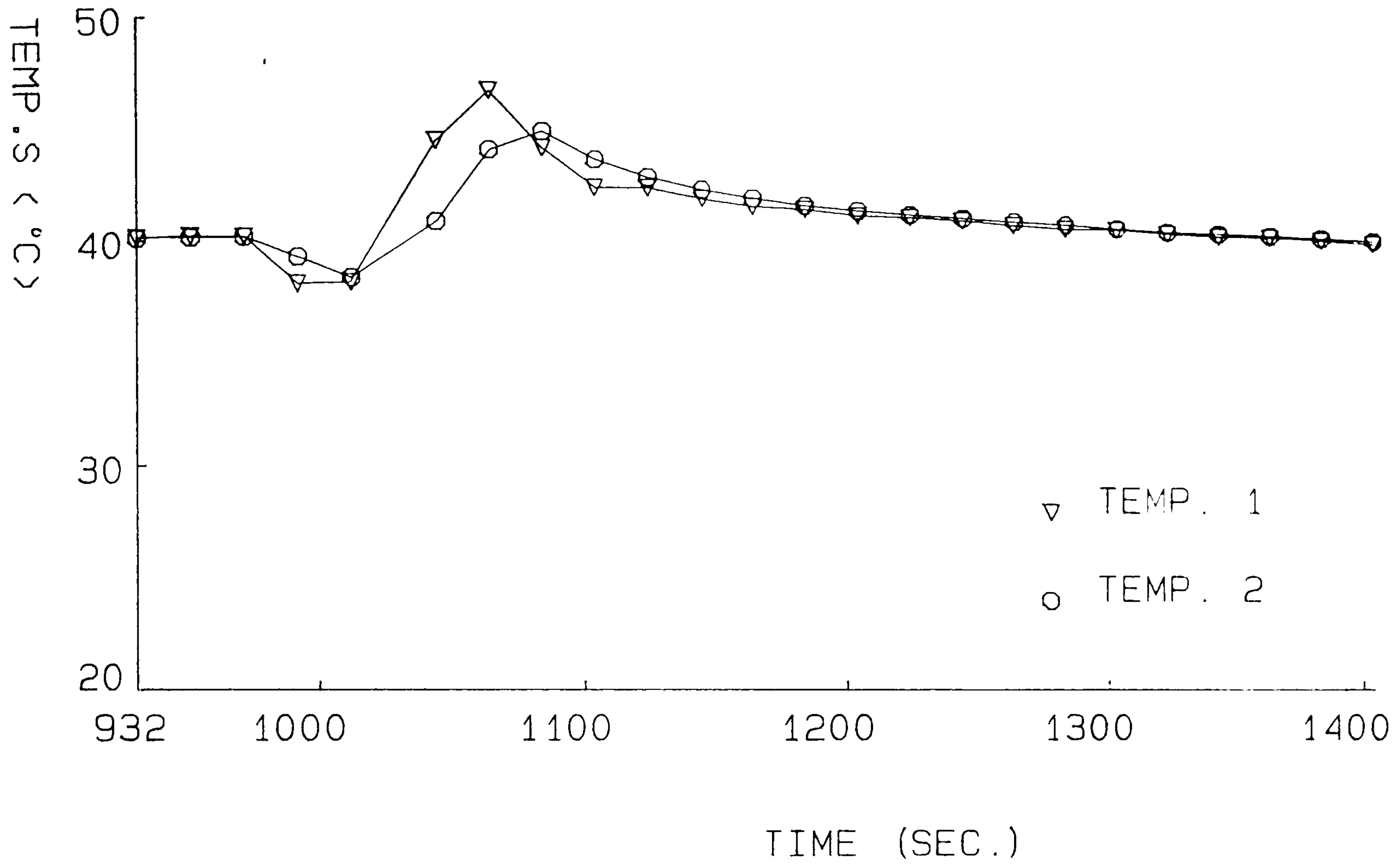


Figure 7.4(b) Temperature measurements for case study No.1 (After learning)

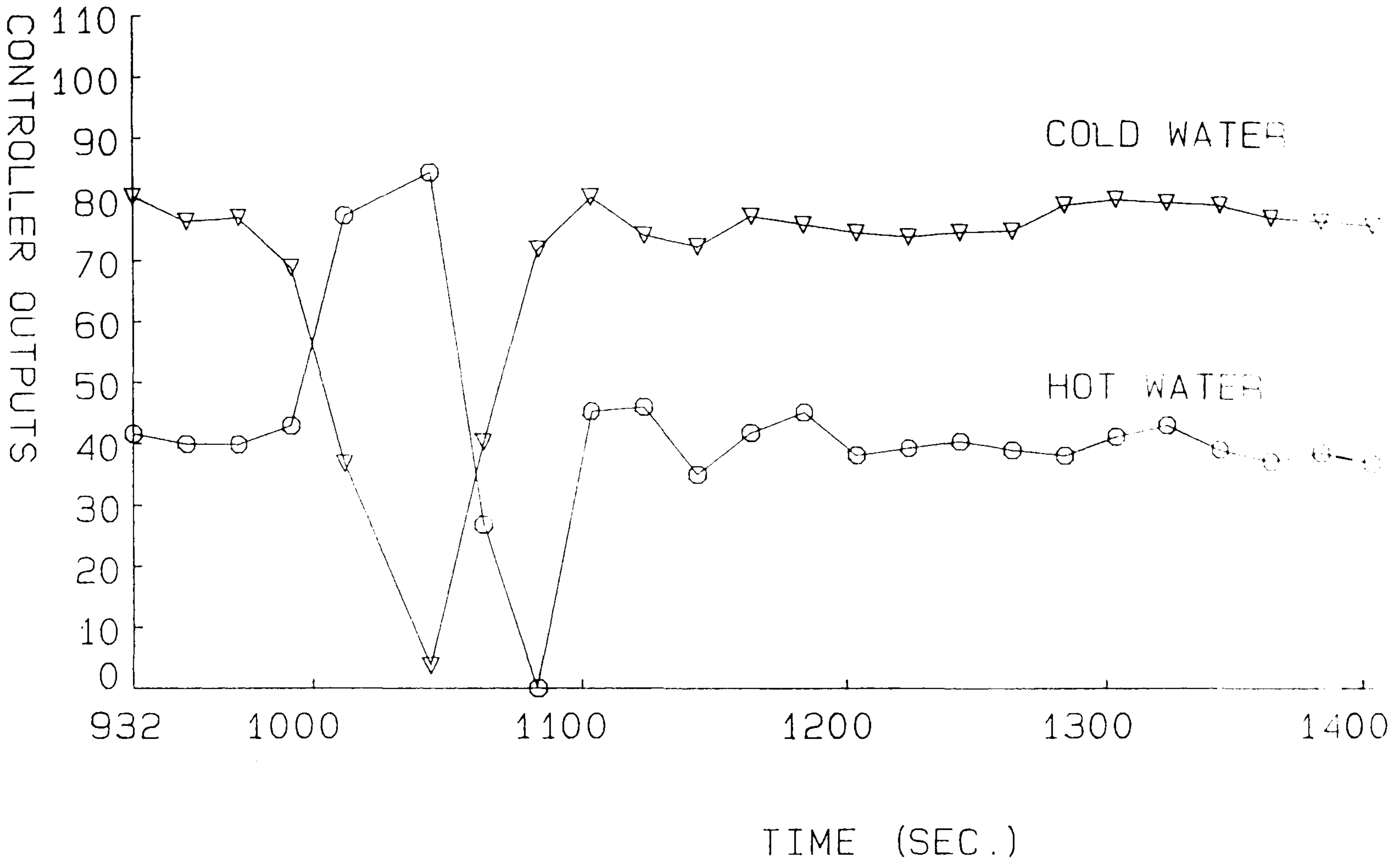


Figure 7.4(c) Controller outputs for case study No.1

(After learning)

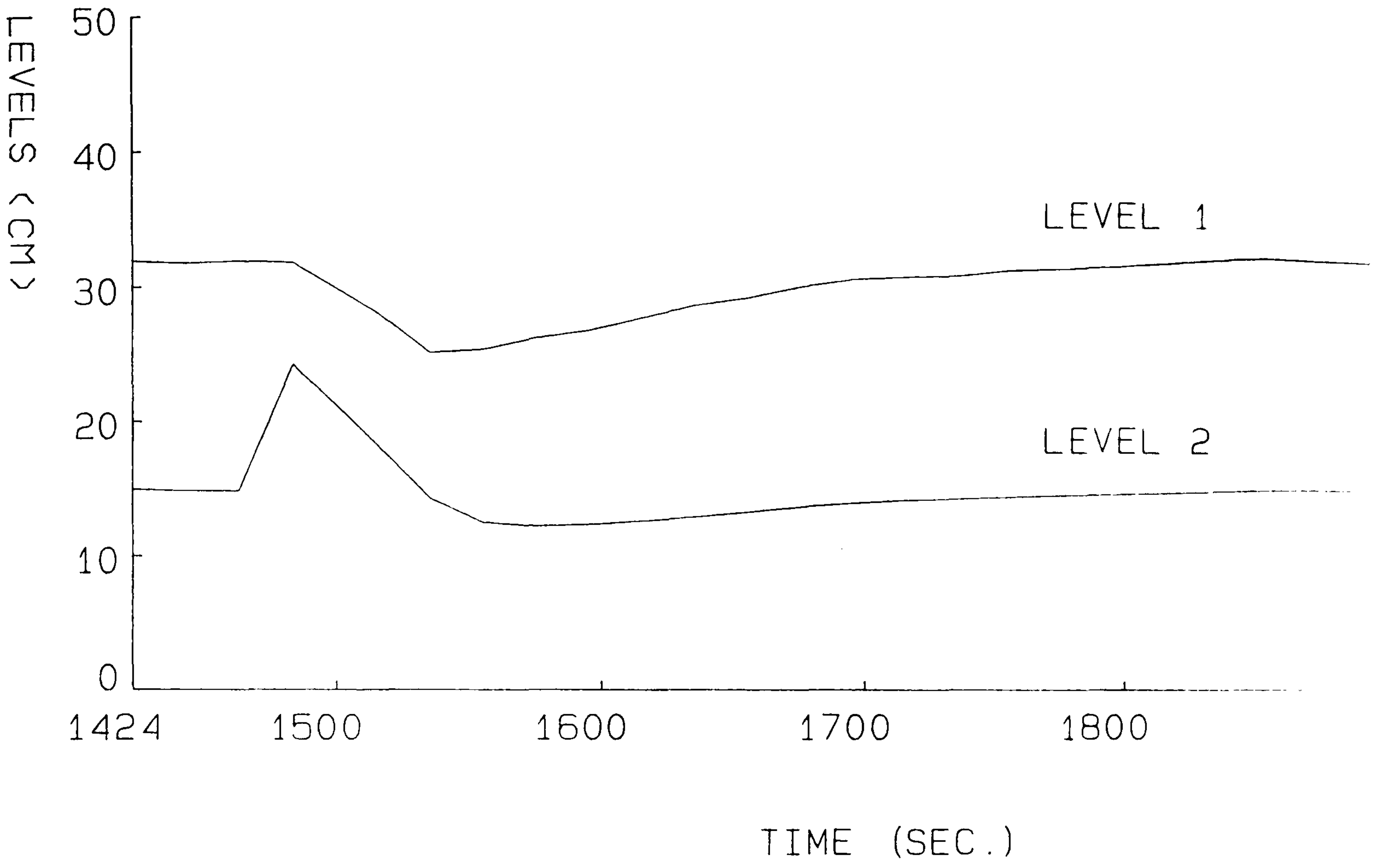


Figure 7.5(a) Level measurements for case study No.2

(Before learning)

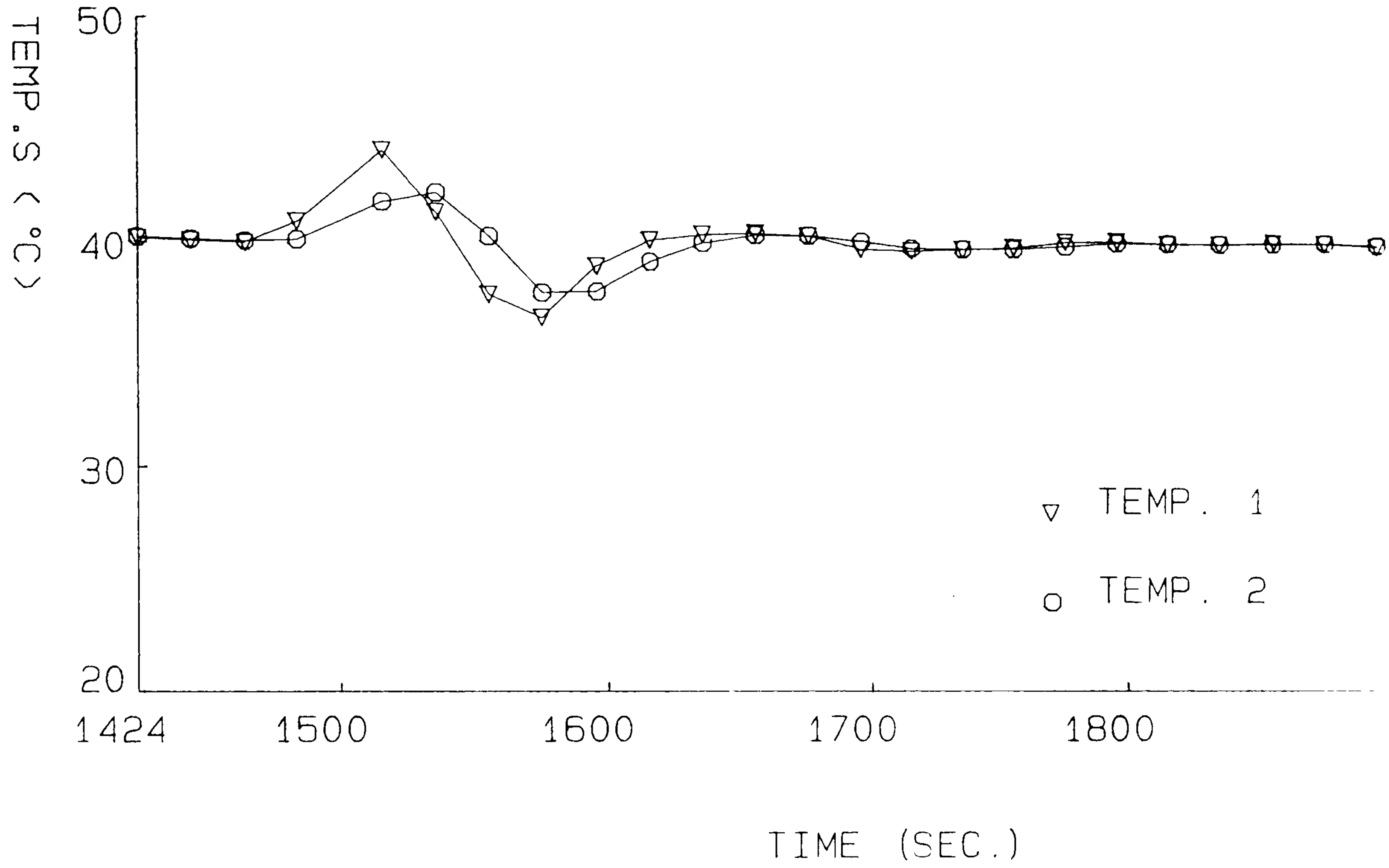


Figure 7.5(b) Temperature measurements for case study No.2 (Before learning)

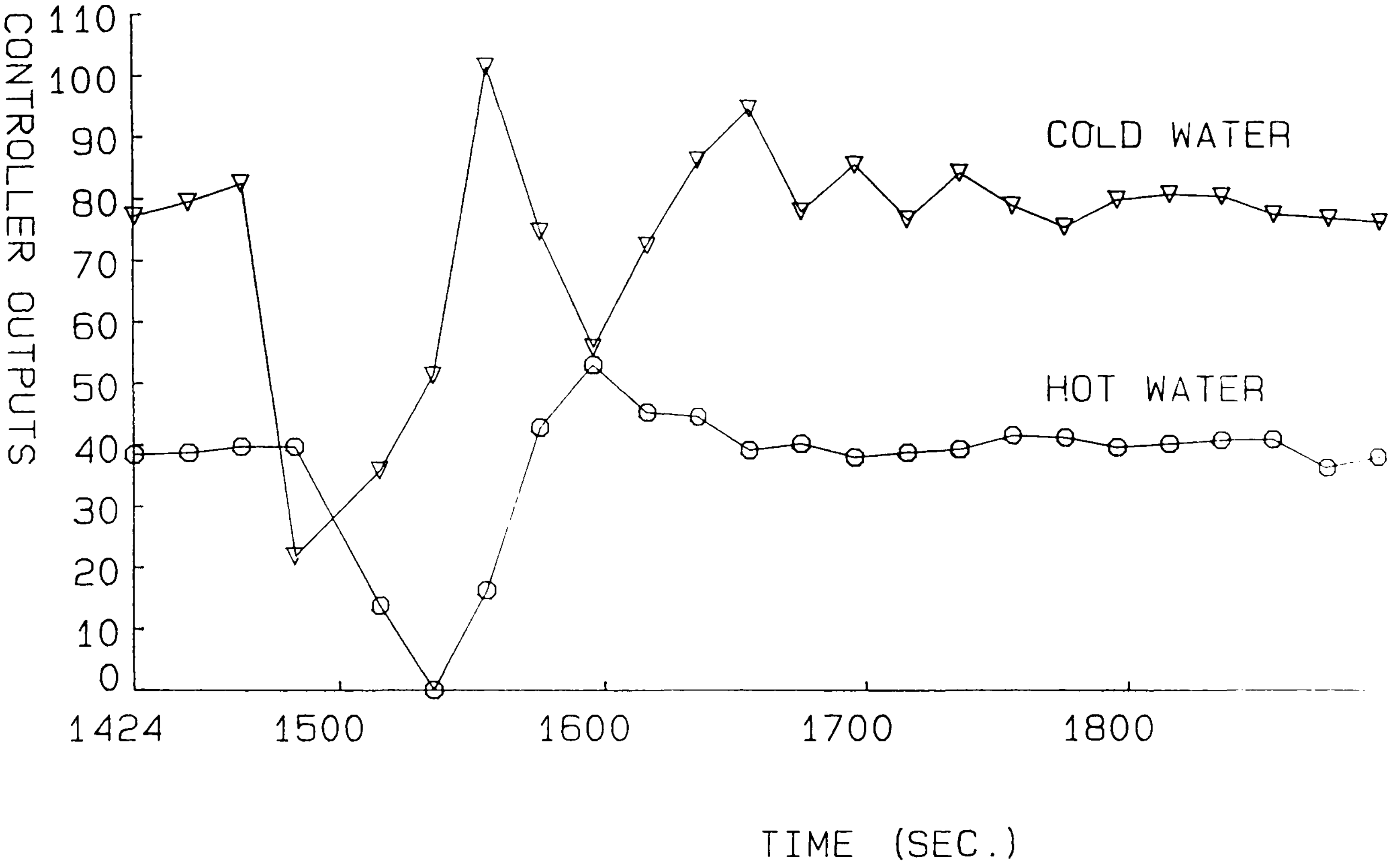


Figure 7.5(c) Controller outputs for case study No.2
(Before learning)

Performance before learning	Performance after learning
Initiated fault: Hand valve 2 blocked Proposed by a heuristic rule? No No. of hypothesis generated & tested: 6 On-line learnt rule: IF Level in tank 1 steady Level in tank 2 increase Cold water flow decrease Hot water flow decrease THEN Hand valve 2 is blocked valve fail	Initiated fault: Hand valve 2 blocked Proposed by a heuristic rule? Yes

Figure 7.6 Performance of the diagnosis system for case study No. 2

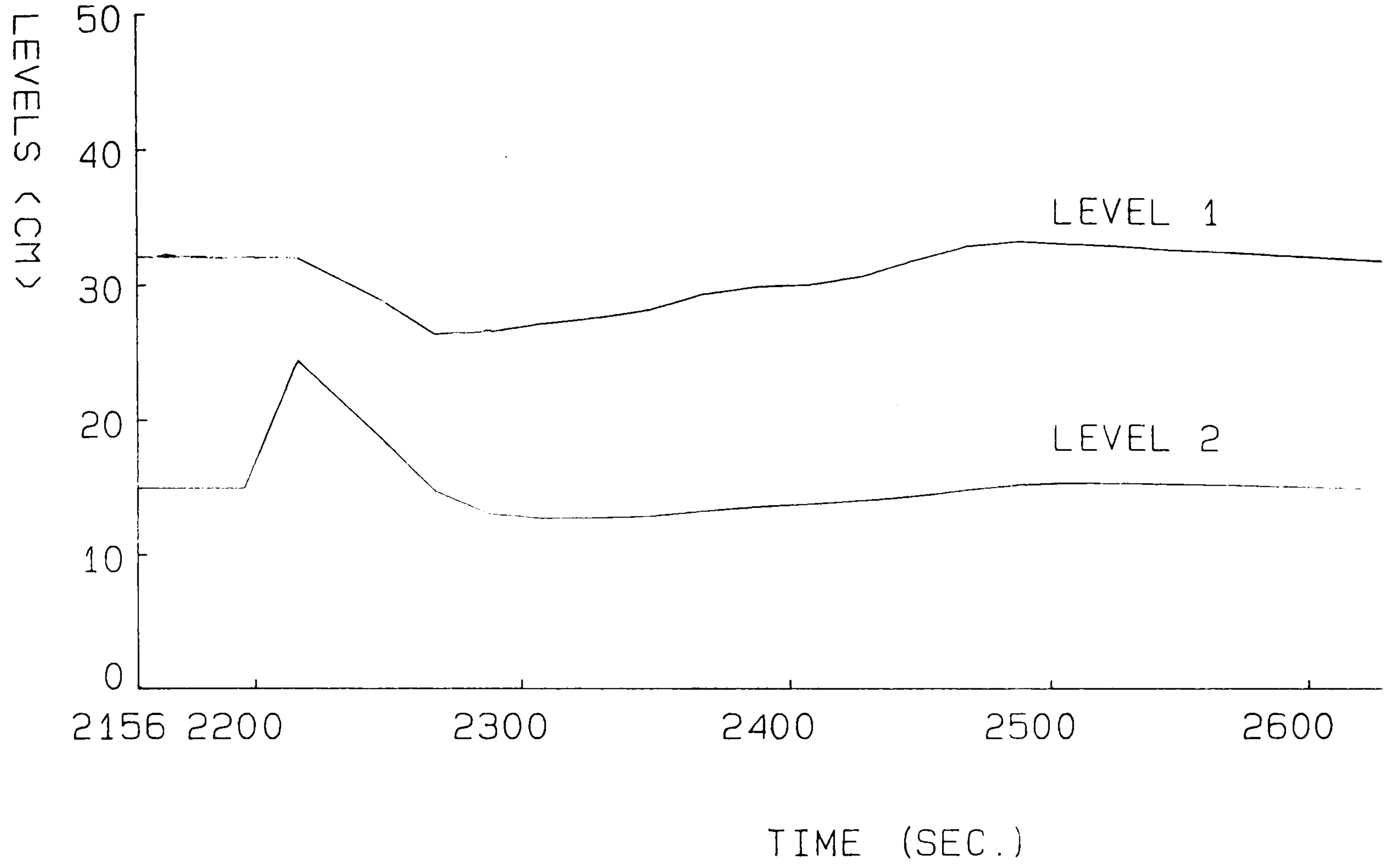


Figure 7.7(a) Level measurements for case study No.2
(After learning)

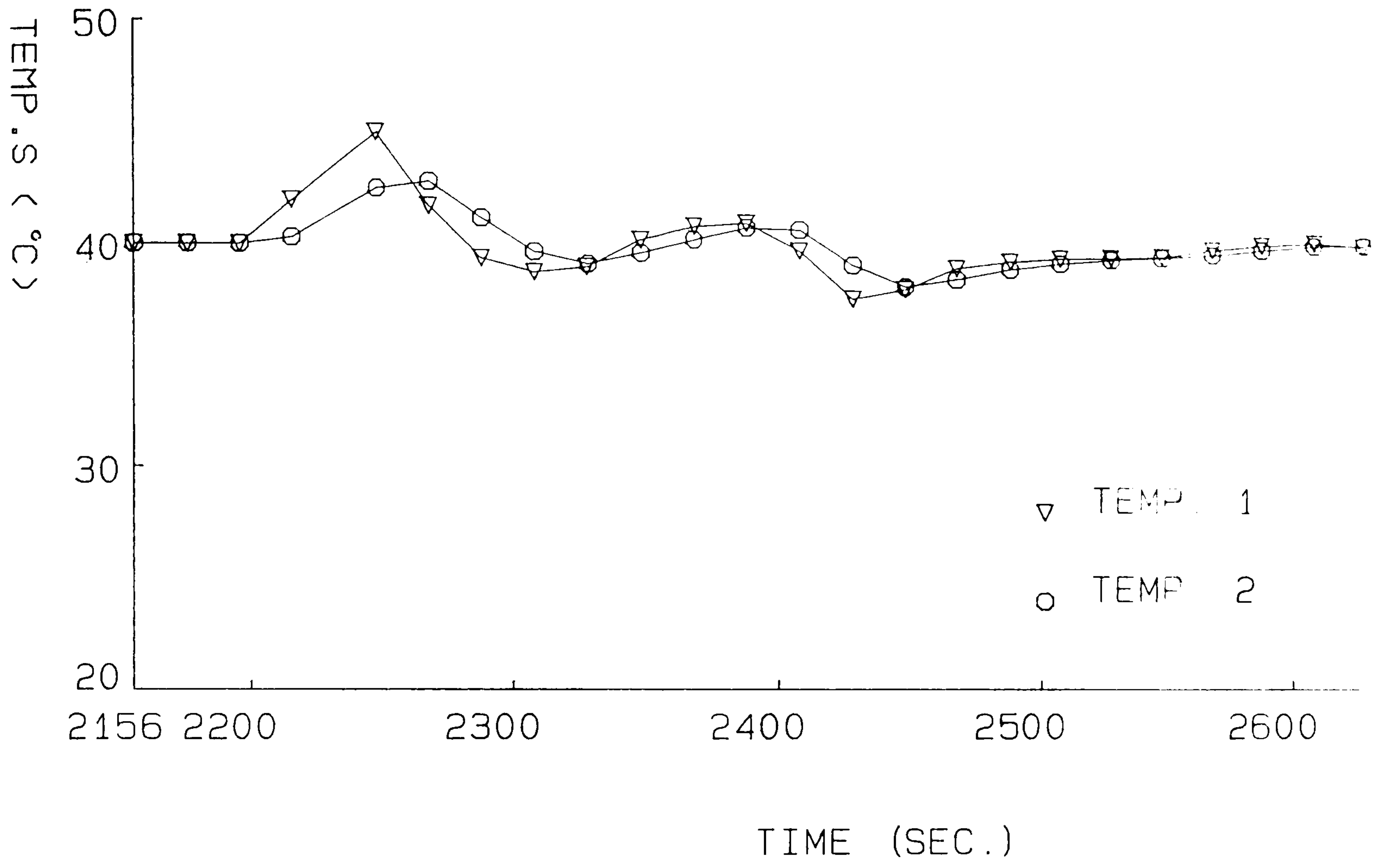


Figure 7.7(b) Temperature measurements for case study No.2 (After learning)

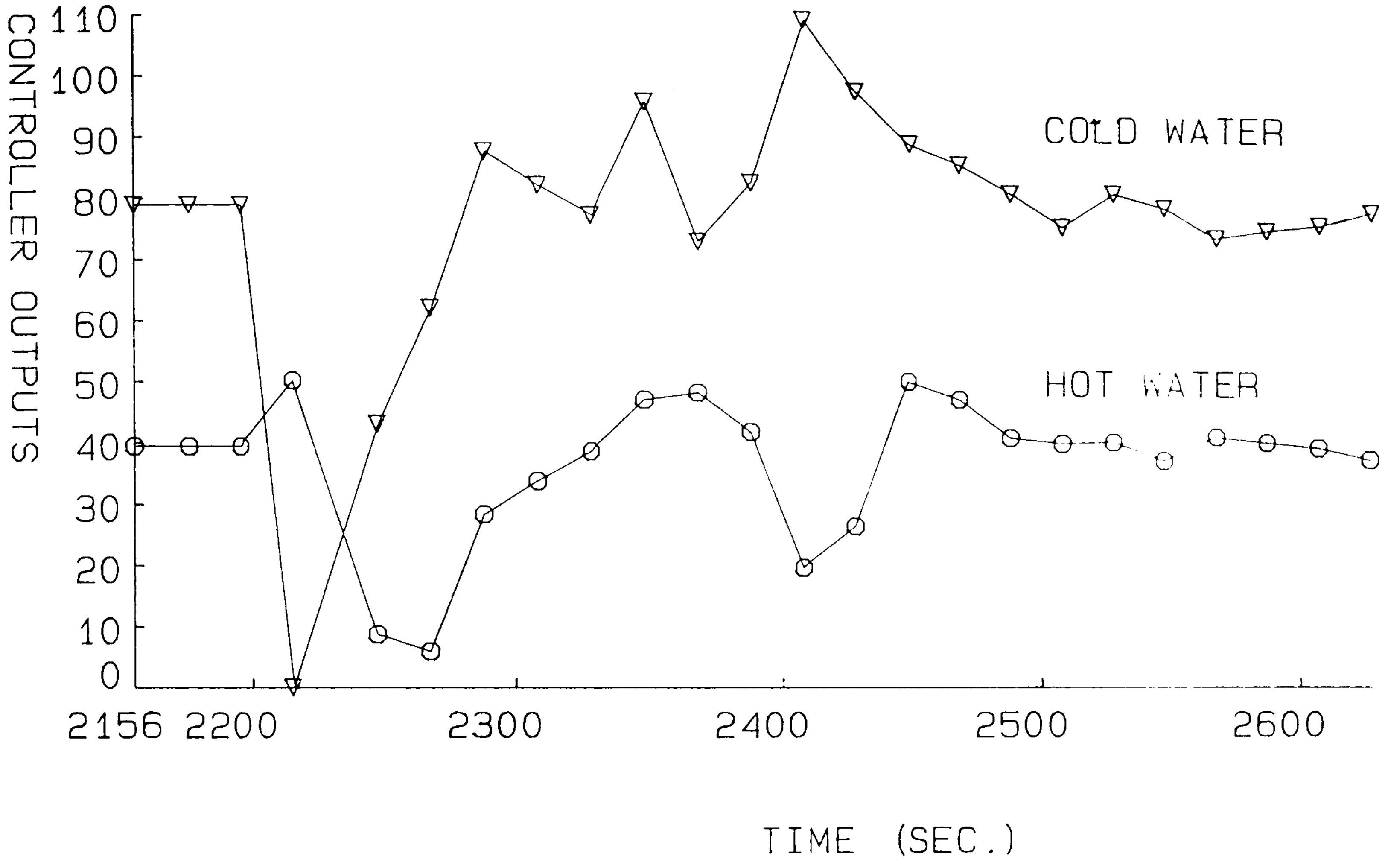


Figure 7.7(c) Controller outputs for case study No.2

(After learning)

Chapter 8

On-line fault diagnosis using neural network techniques

8.1 Introduction

Several different knowledge based fault diagnosis systems have been described in the previous several chapters. The satisfactory performance of these systems suggests that knowledge based expert systems may be used to achieve reliable automated fault diagnosis. However, there are some limitations associated with current knowledge based expert systems. The task of knowledge acquisition is often tedious because a knowledge engineer may often have little knowledge about the operation of a specific process and an experienced process operator may also know little about knowledge engineering, and this issue is referred to as the "knowledge engineering bottle neck" (Moor and Kramer 1986, Price and Lee 1988). This is especially the case for most experience based (or shallow knowledge based) expert systems. The knowledge base of a deep knowledge based expert system contains information on process unit functions, process system structures, as well as a model of the process, and the development of such a knowledge base is also time consuming.

The performance of knowledge based diagnosis systems is affected by the accuracy of their knowledge. The incompleteness in the knowledge base of an expert system could result in potential failures and, furthermore, the performance may not degrade gracefully but could collapse suddenly (Price and Lee 1988). For rule based diagnosis systems, any inaccuracies in the rules can result in a wrong diagnosis (Pazzani 1986, 1987, Rich and Venkatasubramanian 1989). As demonstrated in Chapter 6, the inappropriate parameters in deep knowledge based diagnosis systems can also result in poor performance. The performance of any on-line diagnosis systems is also affected by measurement noise and, therefore, it is important to pre-process the raw data from on-line measurements (Doraiswam and Jiang 1989).

Taking account of these limitations, an alternative approach, which uses neural network techniques (Vemuri 1988, Wasserman 1989, Aleksander and Morton 1990) to diagnose process faults, is investigated in this chapter. The idea for investigating such an approach has its root in the work presented in the previous chapter. In investigating self-learning of heuristic rules described in the previous chapter, the good learning property of neural networks is recognised and it is realised that the data used in learning heuristic rules can also be used to train a neural network. A multi-layer feed forward neural network is established and is trained by symptom-fault pairs obtained from past experience or from simulation analysis. The neural network can abstract the relations between symptoms and faults in the training data and store this information as the trained network weights and, therefore, the trained network can then be used to diagnose faults. With the properties of learning, generalisation, and abstraction (Wasserman 1989), a neural network based diagnosis system can overcome some of the limitations of current knowledge based systems. Furthermore, it is easy to develop and performs in a robust manner. Several neural network based diagnosis systems have been reported recently. Dietz *et al* (1989) developed a real-time fault diagnosis system for jet and rocket engines using neural networks. Venkatasubramanian and Chan (1989) describe a neural network based diagnosis system for a fluidized catalytic cracking unit. Watanabe *et al* (1989) investigate the use of neural networks to diagnose incipient faults in chemical processes. In this chapter neural network based on-line fault diagnosis systems are developed for the pilot scale mixing process and the continuously stirred tank reactor (CSTR) system. The performance of the systems is investigated under partial information and under partially incorrect information. The feasibility of applying a neural network based diagnosis system, developed using simulation data, to a real process is also demonstrated.

The chapter is organised as follows: the next section briefly describes some fundamental concepts in neural networks and the backpropagation training technique. Section 8.3 describes the architecture of a neural network based on-line fault diagnosis system, and the application of such a diagnosis system to the pilot scale mixing process is presented in Section 8.4, where the performance of the system is investigated through a series of experiments. Section 8.5 presents a neural network based diagnosis system for the CSTR system, which is more complicated than the mixing process, and the last section contains some concluding remarks.

8.2 Neural network techniques

This section presents a brief introduction to neural network techniques, detailed descriptions can be found in the literature (Vemuri 1988, Wasserman 1989, Aleksander and Morton 1990).

8.2.1 The artificial neuron

An artificial neuron, which intends to mimic the function of a human neuron, is shown in Figure 8.1, where a set of inputs, $X_1, X_2, \cdots, X_n$, are multiplied by their corresponding weights, $W_1, W_2, \cdots, W_n$, and summed in the neuron to produce the signal, NET. The output of the neuron, OUT, is obtained by applying NET to an activation function, F. The computation of NET and OUT is listed as follows:

$$NET = X_1 W_1 + X_2 W_2 + \cdots + X_n W_n \tag{8.1}$$

$$OUT = F(NET) \tag{8.2}$$

A commonly used activation function is

$$OUT = \frac{1}{1 + e^{-NET}} \tag{8.3}$$

which is known as the logistic function or "Sigmoid" (Vemuri 1988, Wasserman 1989).

8.2.2 Artificial neural networks

A typical artificial neural network is shown in Figure 8.2, where neurons are organised into several layers. The output of each neuron is connected to the inputs of all the neurons in the successive layer through corresponding weights. The layer which accepts inputs from the outside world is called the input layer, while the layer which provides outputs to the outside world is called the output layer, and the other layers are called hidden layers. There is only one hidden layer in Figure 8.2. The outputs of the network are affected by both its inputs and its weights.

There are two main operations in the use of neural networks: training (learning) and generalisation. The process of training can be divided into supervised training and unsupervised training. In supervised training, the network is provided with input vectors and corresponding target vectors, which are called training pairs.

An input vector is applied and the output vector of the network is calculated and compared with the corresponding target vector. The difference (error) is fed back through the network and the weights are adjusted according to an algorithm which tends to minimise the error. The input vectors in the training set are applied sequentially and the errors are calcuated and weights adjusted for each training pair, until the error for the entire training set is at an acceptable level. Unsupervised training, on the other hand, requires no target vectors for the outputs. The training algorithm modifies the network weights to produce output vectors that are consistent, in that both the application of one of the training vectors and the application of a vector that is sufficiently similar to it will produce the same pattern of outputs. On completion of training the network can operate in a generalisation phase where it produces outputs for similar or novel input patterns.

8.2.3 Backpropagation training

In this chapter the algorithm, used in developing neural network based diagnosis systems, is the backpropagation training algorithm (Lippmann 1987, Wasserman 1989), which belongs to the category of supervised training. The backpropagation training process contains a "forward pass" phase, in which an input vector is applied to the network to produce an output vector, and a "reverse pass" phase, in which the differences between targets and outputs are calculated and the network weights are adjusted to minimise the differences. The weights are adjusted by the following algorithm which minimizes the squared errors.

$$\Delta W_{pq,k}(n+1) = \eta(\delta_{q,k} OUT_{p,j}) + \alpha[\Delta W_{pq,k}(n)] \tag{8.4}$$

$$W_{pq,k}(n+1) = W_{pq,k}(n) + \Delta W_{pq,k}(n+1) \tag{8.5}$$

where $W_{pq,k}(n)$ is the value of the weight from neuron p in the jth layer to neuron q in the next layer (kth layer) at step n (before adjustment), $W_{pq,k}(n+1)$ is the weight at step $n+1$ (after adjustment), $\Delta W_{pq,k}(n+1)$ is the adjustment in weight, $OUT_{p,j}$ is the value of OUT for neuron p in the jth layer, $\delta_{q,k}$ is a common factor in the gradient of the squared error, η is the training rate coefficient, and α is the momentum coefficient. For output-layer neurons (if the kth layer is the output layer),

$$\delta_{q,k} = OUT_{q,k}(1 - OUT_{q,k})(Target_q - OUT_{q,k}) \tag{8.6}$$

where $Target_q$ is the qth element of the target vector corresponding to the qth element of the output vector, $OUT_{q,k}$. Finally, for hidden layer neurons (if the kth

layer is a hidden layer),

$$\delta_{q,k} = OUT_{q,k}(1 - OUT_{q,k})(\sum_{r} \delta_{r,l} W_{qr,l}) \tag{8.7}$$

where $W_{qr,l}$ is the weight from neuron q in the kth layer to neuron r in the next layer (the lth layer). Eq(8.7) recursively determines the δ values for each hidden layer.

8.3 On-line process fault diagnosis using neural networks

8.3.1 System structure

The proposed fault diagnosis system is based on the fact that a neural network can learn. The training pairs used are a set of symptoms and the corresponding faults. After training, the neural network will determine the relationship between a specific symptom and the corresponding fault, and store this information as the trained weights. Since the information about the monitored process is obtained through on-line measurements, the symptoms are represented by on-line measurements. The proposed neural network based diagnosis system is shown in Figure 8.3. The on-line measurements are processed, for example scaled, by an information pre-processor into a suitable form which can be directly applied to the network. This processed information is known as the "symptom vector", S, and the outputs of the network indicate the diagnosis result and is termed the "diagnosis vector", D.

The training data can be obtained from past experience or from simulation analysis. Ishida (1988) describes a method for the automatic generation of diagnostic rules through qualitative simulation. Instead of generating rules, the simulation data can also be used to train a neural network.

8.3.2 Operation of neural network based diagnosis systems

There are two kinds of operations: training and generalisation. Training is done off-line while generalisation is performed on-line. During the training phase, a set of symptom-fault pairs are applied to the network, and the network weights are adjusted by the backpropagation training algorithm. The training time is affected by the network structure, training parameters, and the number of training pairs, and may take a long time and, therefore, it is performed off-line. Once a network is trained it is ready to be used for diagnosis. When abnormalities occur in the on-line measurements, the information pre-processor will process the measurements and produces a symptom vector which is then applied to the trained network, and the diagnosis result is presented by the diagnosis vector. The generalisation phase can then be performed in a sufficiently short time for implementation on-line. The network can also be re-trained when the new training data are available to improve its performance.

8.4 Neural network based on-line diagnosis of the mixing process

8.4.1 Neural network based fault diagnosis

The above described neural network based on-line diagnosis technique has been applied to the pilot scale mixing process.

The information pre-processor

The information pre-processor shown in Figure 8.3 for this application is a quantitative to qualitative value converter, which converts the quantitative increments in measurements and controller outputs into their qualitative forms: increase, steady, and decrease. The reason for employing such an information pre-processor is that several qualitative model based diagnosis systems have been developed for the mixing process, and from which the training data for the neural network can be obtained, where the elements of the symptom vector are qualitative deviations in measurements and manipulated variables. Here the numbers 0, 1, 2, and 3 are used to represent information unavailable, decrease, steady, and increase respectively. The assumption that some information may be unavailable has practical meaning. For example, during operation, some sensors may be out of service and, therefore, the information from these sensors is unavailable.

The network structure

The available on-line information on the mixing process are four measurements and two controller outputs, which determines that there should be six neurons in the input layer, each corresponding to a particular information source. The possible

Table 8.1: Training steps for two networks

α	η	training steps (1 hidden layer)	training steps (2 hidden layers)
0.9	0.6	unconverge	unconverge
0.9	0.1	550	1681
0.8	0.5	274	unconverge
0.8	0.4	380	907
0.8	0.2	534	2240
0.6	0.9	828	1361
0.6	0.2	1324	3898
0.5	0.4	1131	2297
0.5	0.2	1796	5215

faults that may occur are considered to be: sensor failures, hand valve 1 is blocked, hand valve 2 is blocked, cold water control valve fails and gives a high output, cold water control valve fails and gives a low output, hot water control valves fails and gives a high output, and hot water control valve fails and gives a low output. Since sensor failures may be present in several forms and do not result in relatively fixed symptoms, at this stage, only the other six failures are considered, which determines that there should be six output-layer neurons. Each output-layer neuron corresponding to a particular fault and its output lies in the range (0,1). When its output is close to 1, it indicates that the corresponding fault has occurred. This output can be taken as an approximate measure of the possibility that a fault has occurred, and only those faults with more than 60% possibility are accepted.

A single hidden layer with five neurons in this network has been chosen. Two hidden layers, each with five neurons, have also been investigated. The training steps required for the two networks to be trained to 95% accuracy under different training parameters are listed in Table 8.1, from which it can be seen that the network with only one hidden layer can be trained very quickly. Therefore, the single hidden layer network has been adopted.

Table 8.2: Training data for the neural network based diagnosis system for the mixing process

Training pairs							Faults
S:	1	1	1	1	3	3	Hot water control
T:	0	0	0	0	0	1	valve fails low
S:	2	2	1	1	3	1	Cold water control
T:	0	0	0	0	1	0	valve fails low
S:	2	2	3	1	3	2	Hand valve 1
T:	0	0	0	1	0	0	is blocked
S:	3	2	2	3	1	2	Hand valve 2
T:	0	0	1	0	0	0	is blocked
S:	3	3	3	2	1	1	Hot water control
T:	0	1	0	0	0	0	valve fails high
S:	2	2	3	3	1	3	Cold water control
T:	1	0	0	0	0	0	valve fails high
S:	2	2	2	2	2	2	
T:	0	0	0	0	0	0	No fault

8.4.2 Network training

Training data

The training data has been obtained from the diagnosis system described in Chapter 5, by inserting a fault to the process model and recording the resulting deviations in simulated measurements and controller outputs. The complete training data is listed in Table 8.2, where S and T are the symptom and target vectors respectively. The elements of S: $s1, s2, \cdots, s6$, are qualitative deviations of temperatures in tank 1 and tank 2, levels in tank 1 and tank 2, and cold and hot water control valve openings respectively. Each element of a target vector corresponds to a specific fault and can take the values of either 1 or 0, with 1 representing the occurrence of the corresponding fault and 0 for no occurrence.

Training parameters

The learning rate parameter, η , was set at 0.8, the momentum coefficient, α, was set at 0.5, and the initial weights were assigned to small uniformly-distributed random values between -0.1 and 0.1. The stopping criterion used for the training process is that the largest error in the error space is less than 5%.

An experiment has been performed in which the network was trained under different parameters, and the results are listed in Table 8.1. It can be seen that smooth adjustment of weights (relatively large α and relatively small η) may provide fast training.

8.4.3 Performance of the neural network based diagnosis system

The trained network has been tested on a set of incomplete and partially incorrect symptoms, in which some elements in the symptom vector were different from their corresponding items in the training data. These partially incorrect symptoms may be due to measurement noise, or some inappropriate parameters in the information pre-processor as described in Chapter 6. If the training data are obtained from simulation analysis, then any model-plant mismatch may also result in these incorrect symptoms.

The symptoms and the diagnosis results are shown in Table 8.3, where the incorrect elements in the symptom vector are marked with "*", and the unavailable elements are marked with "?". It can be seen that the neural network based diagnosis system under partially incorrect and partially unavailable information performs well. One explanation for the good performance could be that the neural network has the ability of abstraction in that it can extract the essential features in the training data. Therefore, when some new data, resembling the training data to some extent, is applied to the neural network, the network can classify the data into appropriate categories.

The network trained on the simulation data has been applied to the real process and the results are also very satisfactory. For instance, consider the following example. A hot water control valve failure giving a low output was initiated by turning off the hand valve in series with the valve (see Figure 3.1). The measurements covering this event are shown in Figure 8.4. The diagnosis system observed that the temperature measurements were abnormal at 440 seconds, when it swiftly collected another four sets of measurements to eliminate measurement noise. The abnormality was presented in all the five sets of measurements, and then the information

Table 8.3: Performance of the diagnosis system under partial and partially incorrect information

Test No.	Symptom vectors, Diagnosis vectors, Faults						
1	S:	$0^?$	1	1	1	3	2^*
	D:	0.0108	0.0002	0.0013	0.1356	0.0481	0.7257
	Fault: Hot water control valve fails low						
2	S:	2	2	$0^?$	2^*	3	1
	D:	0.0001	0.0773	0.0863	0.0043	0.8554	0.0075
	Fault: Cold water control valve fails low						
3	S:	2	1^*	3	1	3	1^*
	D:	0.0177	0.0315	0.0000	0.9412	0.0252	0.0109
	Fault: Hand valve 1 is blocked						
4	S:	2^*	2	2	3	1	1^*
	D:	0.0179	0.2686	0.6050	0.0005	0.0287	0.0013
	Fault: Hand valve 2 is blocked						
5	S:	3	3	2^*	2	$0^?$	1
	D:	0.0051	0.8508	0.1028	0.0041	0.0791	0.0001
	Fault: Hot water control valve fails high						
6	S:	$0^?$	2	3	2^*	1	3
	D:	0.8003	0.0012	0.0007	0.5242	0.0001	0.2958
	Fault: Cold water control valve fails high						

pre-processor calculated the symptom vector as

$$S^T = \begin{pmatrix} 1 & 1 & 1 & 2 & 3 & 3 \end{pmatrix}.$$

Comparing this with the first training pair in Table 8.2, it is observed that the 4th element is different from its counterpart in the training data. The diagnosis result for this symptom is

$$D^T = \begin{pmatrix} 0.0343 & 0.0000 & 0.0127 & 0.0264 & 0.0186 & 0.9641 \end{pmatrix},$$

which indicates that the failure, hot water control valve fails giving a low output, has occurred with a high possibility. It can be seen that the diagnosis result is very accurate.

Several faults have been tested in a similar way as in the above example. The symptom vectors, obtained from on-line measurements, and the diagnosis results are shown in Table 8.4. The elements in the symptom vectors, which are different from their counterparts in the training pairs, are marked with "*". It can be seen from Table 8.4 that the network trained by simulation data performs extremely well on the real process. In Tests 1 and 2, the same failure was initiated, and the resulting symptom vectors are different, which may be due to measurement noise or to the operating conditions being different when the failure was initiated. However, the correct diagnosis result has been obtained for both tests, demonstrating the robust nature of the neural network based diagnosis system, in that it can tolerate measurement noise and model-plant mismatch to some extent. In Test 1, the $4th$ element of the symptom vector is different from its counterpart in the training data, and the diagnosis result shows high accuracy (0.9641). In Test 2, in addition to the $4th$ element, the $3rd$ element is also different from its counterpart in the training data and, in this case, the diagnosis accuracy drops down a little bit (0.8769). This demonstrates the graceful degradation in the performance of neural network based diagnosis systems.

8.5 Neural network based diagnosis of a CSTR system

8.5.1 Neural network based diagnosis

A neural network based diagnosis system is also developed for the CSTR system. Since a qualitative model based diagnosis system has been developed for the CSTR

Table 8.4: Performance of the diagnosis system on the real process

Test No.	Symptom vectors, Diagnosis vectors, Faults						
1	*S*:	1	1	1	2*	3	3
	D:	0.0343	0.0000	0.0127	0.0264	0.0186	0.9641
	Fault: Hot water control valve fails low						
2	*S*:	1	1	2*	2*	3	3
	D:	0.1577	0.0001	0.0027	0.1549	0.0036	0.8769
	Fault: Hot water control valve fails low						
3	*S*:	2	2	1	1	3	1
	D:	0.0000	0.0303	0.0111	0.0333	0.9557	0.0235
	Fault: Cold water control valve fails low						
4	*S*:	2	3*	1	1	3	1
	D:	0.0000	0.0433	0.0163	0.0234	0.9535	0.0159
	Fault: Cold water control valve fails low						
5	*S*:	2	2	3	1	2*	3*
	D:	0.1251	0.0066	0.0000	0.8939	0.0032	0.0542
	Fault: Hand valve 1 is blocked						
6	*S*:	2*	2	2	3	1	2
	D:	0.2032	0.0156	0.6780	0.0003	0.0018	0.0280
	Fault: Hand valve 2 is blocked						

system in Chapter 5, training data can be obtained from that system. The symptoms are the qualitative deviations in on-line measurements and controller outputs. Hence, the information pre-processor used is also a quantitative to qualitative value converter.

There are eleven measurements and three controller outputs, and, therefore, there should be fourteen input-layer neurons, each corresponding to a particular information source. It is assumed that there are eleven possible faults which determines that there should be eleven output-layer neurons, each corresponding to a particular fault. Furthermore, a single hidden layer with ten neurons has been located in the network.

The training data, which is obtained by inserting a fault to the simulated process and recording the resulting qualitative deviations in measurements and controller outputs, is listed in Table 8.5. The elements of the symptom vector, $s1, s2, \cdots, s14$, are the qualitative deviations in the flow rate, temperature, and concentration of external feed reactant, level and temperature in the reactor, flow rate and concentration of the output product, the opening of control valve 1, the recycle flow rate, the flow rate, pressure, and temperature of the cold water to the heat exchanger, and the openings of control valve 3 and control valve 2 respectively. The learning rate coefficient, η, is set to 0.6, the momentum coefficient, α, is set to 0.8, and the initial weights are randomised between -0.1 and 0.1.

Table 8.5: Training data for the neural network based diagnosis system for the CSTR system

Training pairs								Faults
S:	1	2	2	1	3	1	1	
	3	2	3	2	2	2	3	
T:	0	0	0	0	0	0	0	Pipe 1 is blocked
	0	0	0	1				
S:	3	2	2	3	1	3	3	
	1	2	1	2	2	2	1	
T:	0	0	0	0	0	0	0	External feed reactant flow too high
	0	0	1	0				
S:	2	2	2	3	1	1	2	
	1	1	1	2	2	3	1	
T:	0	0	0	0	0	0	0	Pipe 2 or 3 is blocked or pump fails
	0	1	0	0				
S:	2	2	2	3	1	1	3	
	1	2	1	2	2	2	1	Pipe 10 or 11 is blocked or control valve 1 fails low
T:	0	0	0	0	0	0	0	
	1	0	0	0				
S:	2	3	2	2	3	2	2	
	2	2	3	2	2	2	3	
T:	0	0	0	0	0	0	1	External feed reactant temperature abnormal
	0	0	0	0				
S:	2	2	2	2	1	2	2	
	2	2	3	2	2	2	1	
T:	0	0	0	0	0	1	0	Control valve 2 fails high
	0	0	0	0				

(Table 8.5 continued)

Training pairs								Faults
S:	2	2	2	2	3	2	2	
	2	2	1	2	2	2	3	Pipe 7, 8 or 9 is
T:	0	0	0	0	1	0	0	blocked or control
	0	0	0	0				valve 2 fails low
S:	2	2	2	1	3	3	2	
	3	2	3	2	2	2	3	
T:	0	0	0	1	0	0	0	Control valve 1
	0	0	0	0				fails high
S:	2	2	2	2	3	2	2	
	2	1	3	2	2	3	2	Pipe 4, 5, or 6 is
T:	0	0	1	0	0	0	0	blocked or control
	0	0	0	0				valve 3 fails low
S:	2	2	2	2	2	2	2	
	2	3	2	2	2	1	2	
T:	0	1	0	0	0	0	0	Control valve 3
	0	0	0	0				fails high
S:	2	2	1	2	1	2	1	
	2	2	1	2	2	2	1	External feed reactant
T:	1	0	0	0	0	0	0	concentration
	0	0	0	0				too low
S:	2	2	2	2	2	2	2	
	2	2	2	2	2	2	2	
T:	0	0	0	0	0	0	0	No fault
	0	0	0	0				

8.5.2 Performance of the neural network based diagnosis system

After training, the neural network is ready to be used for diagnosis. The trained network has been tested on a set of symptoms which, together with the diagnosis results, are listed in Table 8.6. The symptoms used correspond to the first eleven items in Table 8.5, and are obtained by making some measurements unavailable (marked with "?") and changing some qualitative increments (marked with "*"). It can be seen that satisfactory diagnosis can still be obtained under partially unavailable and partially incorrect symptoms.

The good performance of neural networks based diagnosis systems demonstrates their feasibility in on-line process fault diagnosis. In some cases it would be better to develop a neural network based diagnosis system rather than a rule based diagnosis system. For example, Hanakuma (1989) describes an expert system for fault diagnosis at petrochemical plants, where diagnosis rules are represented by a table, representing relations between conceivable faults and observable symptoms, called the Cause-Symptom Matrix (CSM). The CSM is essentially a set of training pairs and can be used to train a network. The trained network will perform better than a rule based system using rules obtained entirely from the CSM. Ishida (1988) demonstrates that diagnostic rules can be obtained through qualitative simulation. A diagnostic rule is constructed by inserting a fault as a disturbance to the qualitative model of the process, and recording the observable qualitative deviations in measurements. The data obtained from qualitative simulation can also be used to train a network which will then perform even better.

Table 8.6: Performance of the neural network based diagnosis system for the CSTR system

Test No.	Symptom vectors, Diagnosis vectors, Faults							
1	*S*:	1	2	$0^?$	1	3	$0^?$	1
		3	3^*	3	2	3^*	2	3
	SD:	0.0023	0.0113	0.0061	0.0110	0.0053		
		0.0112	0.0325	0.0001	0.0058	0.0000	0.9629	
	Fault: Pipe 1 is blocked							
2	*S*:	3	$0^?$	2	2^*	1	3	3
		1	2	1	3^*	2	$0^?$	1
	SD:	0.0844	0.2207	0.0001	0.1529	0.0002		
		0.3512	0.0003	0.0035	0.0000	0.7242	0.0000	
	Fault: External feed reactant flow too high							
3	*S*:	$0^?$	2	2	3	1	1	1^*
		1	2^*	1	2	2	3	$0^?$
	SD:	0.0231	0.0000	0.0151	0.0000	0.0048		
		0.0113	0.0001	0.0207	0.9369	0.0009	0.0331	
	Fault: Pipe 2 or 3 is blocked or pump fails							
4	*S*:	2	2	2	2^*	1	1	3
		$0^?$	1^*	1	2	3^*	2	1
	SD:	0.0002	0.0041	0.0008	0.0000	0.0013		
		0.0279	0.0445	0.9089	0.0430	0.0106	0.0006	
	Fault: Pipe 10 or 11 is blocked or control valve 1 fails low							
5	*S*:	2	3	2	$0^?$	2^*	2	3^*
		2	2	3	2	3^*	2	3
	SD:	0.0000	0.0658	0.0180	0.1835	0.0001		
		0.2600	0.7801	0.0002	0.0000	0.0009	0.0223	
	Fault: External feed reactant temperature abnormal							

(Table 8.6 continued)

Test No.	Symptom vectors, Diagnosis vectors, Faults							
6	S:	2	2	2	2	$0^?$	2	2
		2	1^*	3	1^*	2	2	$0^?$
	SD:	0.0022	0.0017	0.0189	0.0131	0.0000		
		0.9757	0.0028	0.0023	0.0005	0.0090	0.0032	
	Fault:	Control valve 2 fails high						
7	S:	2	3^*	2	2	3	2	$0^?$
		2	3^*	1	2	$0^?$	2	3
	SD:	0.0254	0.0124	0.0015	0.0117	0.9648		
		0.0000	0.0025	0.0001	0.0026	0.0046	0.0108	
	Fault:	Pipe 7, 8, or 9 is blocked or control valve 2 fails low						
8	S:	2	3^*	1^*	$0^?$	3	3	2
		3	2	3	2	$0^?$	2	3
	SD:	0.0008	0.3924	0.0014	0.9037	0.0019		
		0.0271	0.0890	0.0000	0.0000	0.0188	0.0015	
	Fault:	Control valve 1 fails high						
9	S:	$0^?$	2	2	2	3	2	3^*
		2	1	3	1^*	2	3	2
	SD:	0.0000	0.0000	0.8435	0.0128	0.0061		
		0.0054	0.4570	0.0001	0.0012	0.0072	0.0024	
	Fault:	Pipe 4, 5, or 6 is blocked or control valve 3 fails low						
10	S:	2	$0^?$	2	2	3^*	2	2
		3^*	3	2	2	2	1	2
	SD:	0.0664	0.6525	0.0001	0.2882	0.1443		
		0.0006	0.0041	0.0001	0.0000	0.0007	0.1039	
	Faults:	Control valve 3 fails high						
11	S:	2	2	$0^?$	2	1	2	$0^?$
		2	1^*	1	2	2	2	$0^?$
	SD:	0.9021	0.0024	0.0022	0.3043	0.0012		
		0.1016	0.0000	0.0000	0.0020	0.1281	0.0011	
	Fault:	External feed concentration too low						

8.6 Conclusions

Neural network based on-line fault diagnosis systems have been developed for industrial processes. It is demonstrated that a neural network can acquire diagnostic knowledge from a set of symptom-fault pairs, which may be obtained from experience or from simulation analysis. This is performed automatically by a training algorithm in a much easier manner than the development of a knowledge based diagnosis system. It is also demonstrated that the performance of the neural network based diagnosis systems under partially incorrect and partially unavailable information is very satisfactory. This suggests that neural network based diagnosis systems can tolerate measurement noise and model-plant mismatch to some extent, and shows the feasibility of applying it to on-line process fault diagnosis.

To obtain good performance the training data must cover all the faults that may occur. Neural network based diagnosis systems cannot diagnose faults which are not presented in the training data. On the other hand, a knowledge based diagnosis system may provide some useful guidance when it cannot diagnose a novel fault and, furthermore, knowledge based diagnosis systems often have transparent knowledge bases and can provide good explanation facilities. The combined use of knowledge based expert systems techniques with the neural networks technique could enhance the advantages in both areas and would be a useful future research topic.

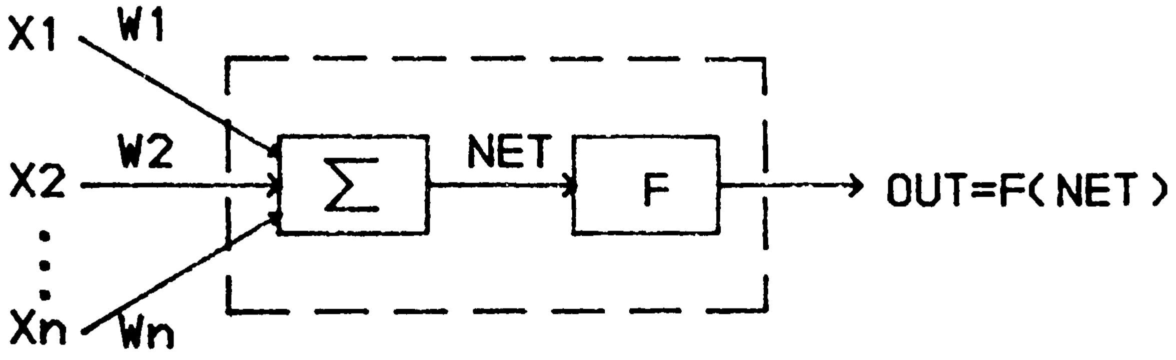


Fig.8.1 An artificial neuron

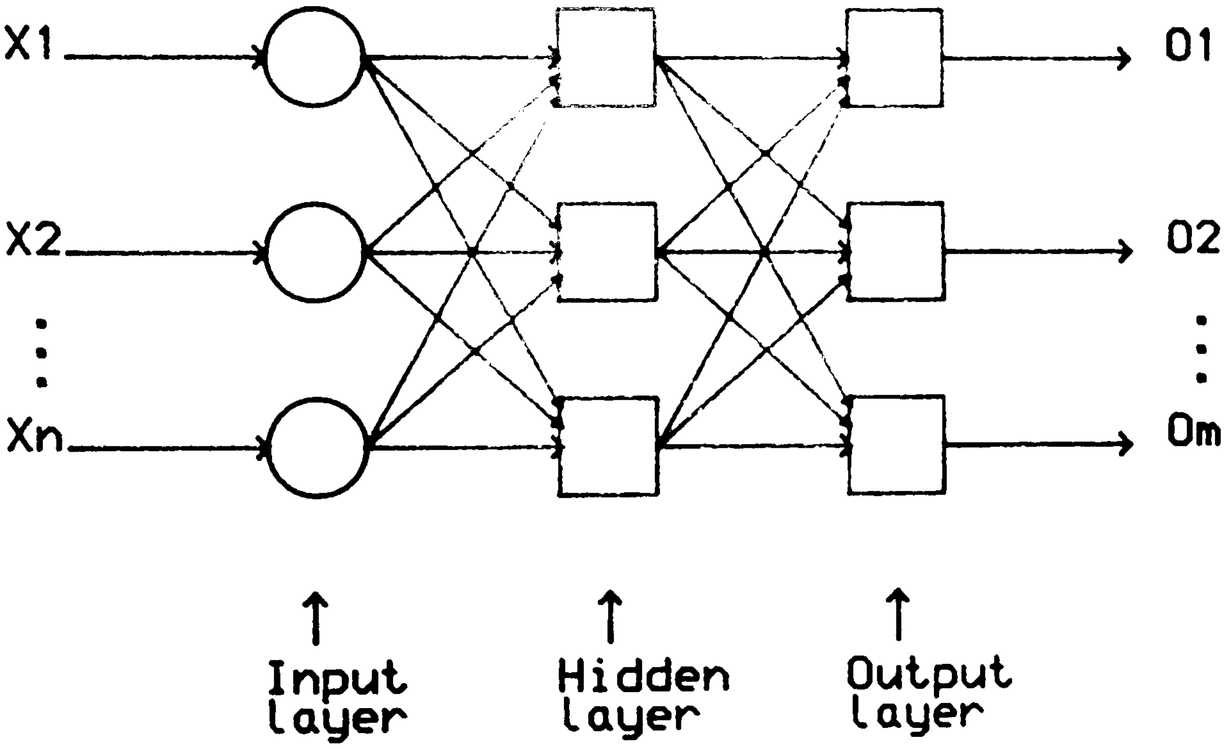


Fig.8.2 An artificial neural network

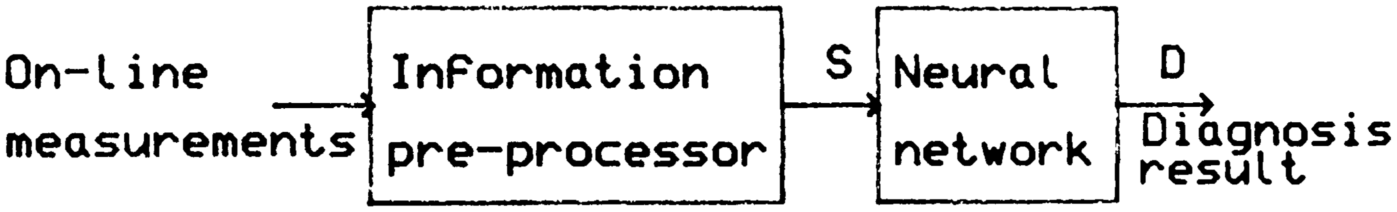


Fig.8.3 Neural network based diagnosis

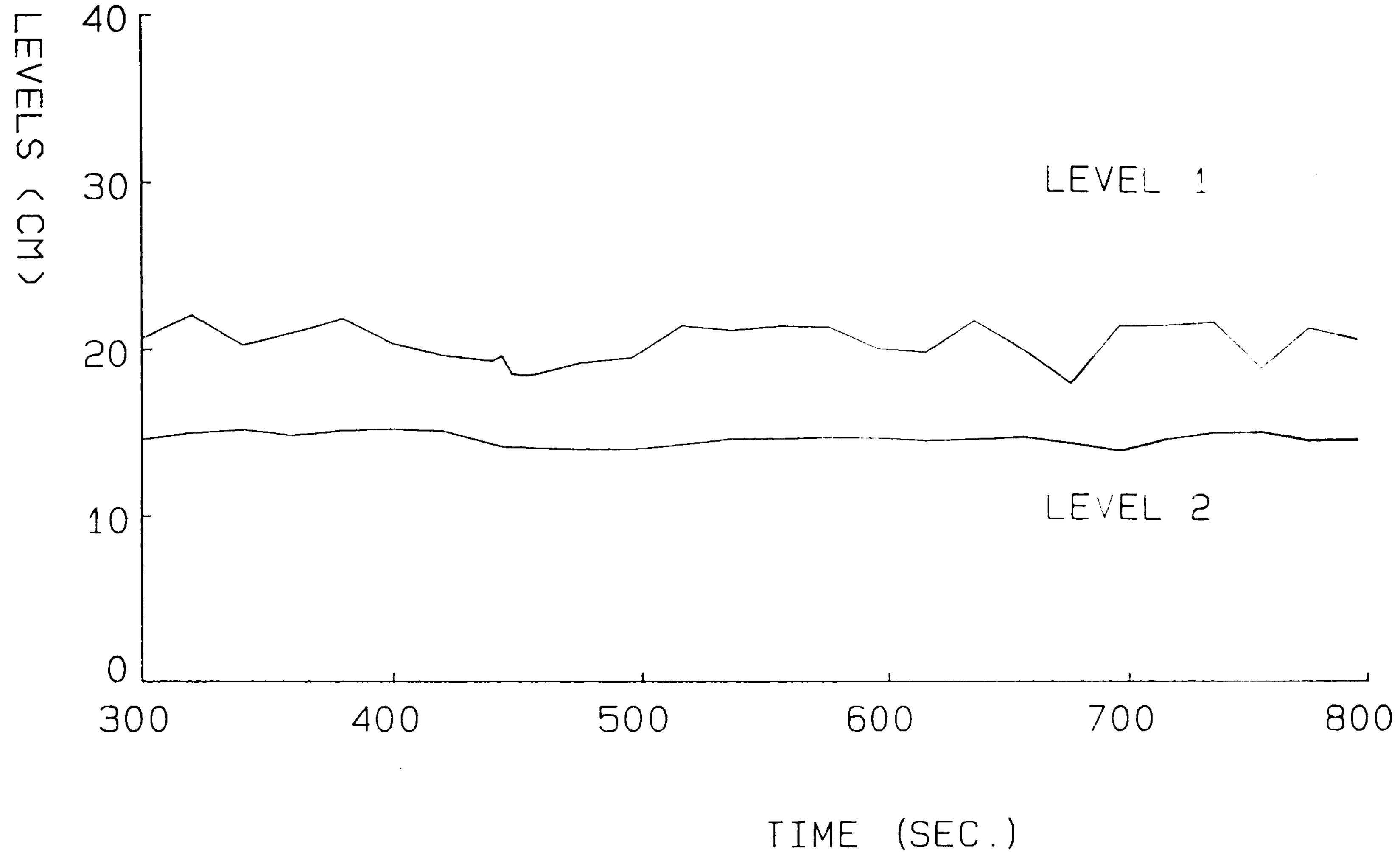


Figure 8.4(a) On-line level measurements

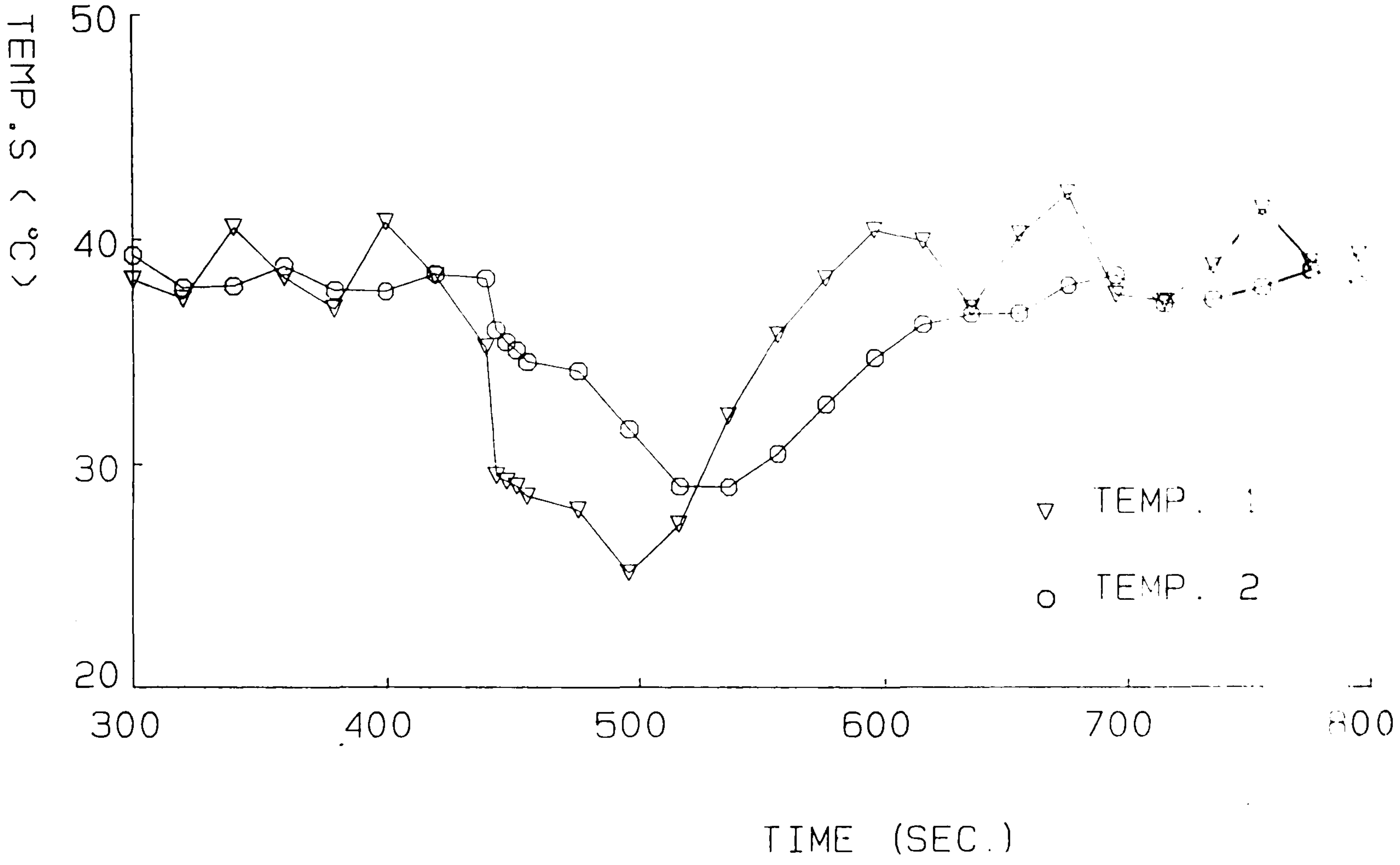


Figure 8.4(b) On-line temperature measurements

Chapter 9

Conclusions and suggestions for future work

The research carried out has been concerned with applications of expert systems techniques to on-line process control and fault diagnosis, and the majority of this research is on knowledge based systems for on-line process fault diagnosis. Several on-line expert systems have been developed and tested. The research results on rule based control demonstrate that rule based controllers are useful in cases where mathematical models of the controlled process cannot be obtained or are very difficult to obtain and, therefore, conventional control algorithms may not be efficiently applied. The research work described in Chapter 3 also suggests that the property of a rule based controller is largely determined by the rules and, therefore, unlike conventional controllers, such as PID controllers, the performance of a rule based controller is not very sensitive to its parameter changes compared with that of a conventional controller.

On-line fault diagnosis is regarded as a supervisory task in this research. Knowledge based systems have more perspectives in performing such supervisory tasks than performing lower level regulatory tasks, since many supervisory tasks cannot be represented by a concise mathematical algorithm.

Several different knowledge based approaches for on-line fault diagnosis have been investigated in this research. The research emphasises deep knowledge based approaches, where the deep knowledge can be qualitative models and/or knowledge on the connectivity and functions of process units. The research work described in Chapter 4 suggests that developing diagnostic rules based on knowledge on system structure and component functions would be a systematic way for developing rule based diagnosis systems. Diagnostic rules developed in such a way could cover a large range of potential faults. Inference based on these rules has higher certainty since these rules capture the underlying first principles of the diagnosed process. Any experimental knowledge can also be integrated with the rules developed from knowledge of system structures and component functions. By decomposing the process being diagnosed into several subsystems, the search for a fault can be conducted very efficiently.

Qualitative modelling provides a means for reasoning based on inaccurate process models and inaccurate measurements. Qualitative reasoning is suitable for the purpose of fault diagnosis for which exact reasoning may not be necessary and, furthermore, the exact severity of a fault is usually not known and to simulate the effect of a fault, qualitative simulation could be more appropriate. The fault diagnosis systems described in Chapter 5 demonstrate that the confluence based qualitative reasoning technique (De Kleer and Brown 1984) is very suitable for process fault diagnosis. The set of confluences for a process can be derived from its mathematical model. Using the confluence representation, various fault models can be easily handled. The effect of a fault can be represented by setting some variables in the qualitative model to certain specified values and, therefore, it is not necessary to have different models for different conditions. To simulate the effect of a fault, it is only necessary to alter some variables in the qualitative model and all the operations based on the model are not changed, that is, it is not required to have different fault simulation procedures for different faults.

The order of magnitude reasoning approach described in Chapter 5 suggests that by using certain available quantitative information, ambiguity in qualitative reasoning can be reduced to a certain extent.

The works described in Chapter 6 and Chapter 7 are based on, and supplement, the qualitative modelling based diagnosis approach described in Chapter 5. The ability of reasoning about its own behaviour could make a knowledge based system more intelligent and autonomous. The self-learning diagnosis system described in Chapter 6 can be understood as a hierarchical diagnosis system, where the lower level fault diagnosis system is an ordinary one, identical to that described in Chapter 5, and the upper level diagnosis system will reason about the lower level one if any undesirable performance occurs there. By such means, any inappropriate parameters in the fault diagnosis system could be found. Therefore, the diagnosis system possesses adaptive properties.

The research presented in Chapter 7 suggests that the combined use of deep knowledge and heuristics could improve both diagnostic efficiency and reliability. Heuristics in the form of rules are efficient to use, and deep knowledge could enhance diagnostic reliability and can cover a wide range of potential faults. By using machine learning techniques, heuristic rules can be automatically acquired, and this eases the knowledge acquisition task. Through self-learning of heuristic rules, the diagnosis system can gradually improve its performance in terms of diagnostic efficiency. The diagnosis systems described in Chapter 6 and Chapter 7 are modifications of those described in Chapter 5 and they can be combined to form a new diagnosis system which can reason its own behaviour, learn diagnostic rules, and enhance diagnostic reliability and efficiency.

Chapter 8 presents a different approach to process fault diagnosis which uses neural network techniques. An advantage of such systems is that they are easy to develop provided that training data are available. Training data could be obtained from past operating experience on a process or from simulation analysis. The research results presented in Chapter 8 suggest that neural network based diagnosis systems could work under incomplete information and partially incorrect information and, therefore, they can tolerate the effect of measurement noise, process disturbances, and model plant mismatch in the case that training data are obtained from simulation analysis. This demonstrates the robustness of neural network based diagnosis systems. A further advantage of the neural network based approach is the parallel nature in neural network operations, which can be ideally implemented with the recently developed parallel processing techniques to meet real-time requirements. The neural network based diagnosis system can also be combined with other diagnosis systems described in this thesis to form a diagnosis system which can diagnose faults based on different methodologies.

The research results presented in this thesis have shown the great potential of knowledge based systems in performing on-line process control tasks including both lower level regulation tasks and higher level supervisory tasks. The on-line fault diagnosis systems described in this thesis have been successfully applied to pilot scale and simulated processes. Further extensions of the applications to industrial scale processes could be investigated in future research. As described in Section 4.2, the on-line diagnosis systems developed in this research will not have any side effects on the monitored process, they could be ready for industrial trial. Further research is needed to explore the perspectives of knowledge based systems, as well as other techniques in the field of artificial intelligence, in process control.

The well recognised learning properties of neural networks can be used in the identification of highly non-linear plants. Neural networks can also be used to act as a controller which will learn the plant dynamics and provide control actions accordingly. The generation of control signals could be commissioned by a knowledge based system, and the knowledge based element and the neural network will collectively form a new type of controller.

The recently emerged topic in artificial intelligence — Genetic Algorithms (GA) (Goldburg 1989), which imitates the process of biological evolution, has shown remarkable performance in optimisation and machine learning. This new technique also has potential perspectives in process control. Genetic algorithms based optimisation could be used in performing optimisation tasks in process control. Genetic algorithms based machine learning can be incorporated into knowledge based systems for process control, and makes them more intelligent and autonomous.

Apart from the knowledge based approaches to process fault diagnosis, there also exist other approaches, such as those based on parameter estimation, state estimation, and filtering. These approaches usually require that a mathematical model of the diagnosed process can be developed and the relations between model parameters, or states, and physical parameters of the process are generally required. The combination of a knowledge based approach and a parameter estimation approach would be a future topic of research. Under such a scheme, parameter estimation, state estimation, and filtering can function as parts of the information pre-processor, which provides the knowledge based element with more information about the process. Such a system could then be sensitive to slight faults, and could also cope with the situations where accurate theoretical modelling is difficult to conduct.

Process supervisory tasks include on-line fault diagnosis and other tasks such as suggesting repairing procedures after a fault has been diagnosed, suggesting different controller structures and control algorithms in cases of occurrences of faults. Knowledge based supervisory could be more important for large scale processes where knowledge based systems can be used to provide intelligent coordinations between subsystems of the process to achieve overall profit and to provide alternative control configurations in case of abnormal operating conditions. These tasks which are not investigated in this research could be investigated in future researches. The methodology for performing these tasks could be quite process specific.

References

Aleksander, I. and H. Morton, *An Introduction to Neural Computing,* Chapman and Hall, 1990.

Åström, K. J., "Toward Intelligent Control", *IEEE Control Systems Magazine,* April 1989, PP60–64.

Åström. K. J., J. J. Anton, and K-E. Årzén, "Expert Control", *Automatica,* Vol.22, No.3, 1986, PP277–286.

Bobrow, D. G. (ed), *Artificial Intelligence, Special Volume on Qualitative Reasoning about Physical Systems,* Vol.24, Numbers 1–3, December 1984.

Cooper, D. J., "An Expert System Approach to Process Identification and Adaptive Control", *Knowledge Based Expert Systems for Engineering: Classification, Education and Control,* (eds) P. Sriram and R. A. Adey, Computational Mechanics Publications, 1987, PP29–42.

Davis, R., "Diagnosis Via Causal Reasoning: Paths of Interaction and the Locality Principle", *AAAI-83,* Washington D.C., Aug. 83, PP88–94.

Davis, R., "Reasoning from First Principles in Electronic Troubleshooting", *Developments in Expert systems,* (ed, M. J. Coombs), Academic Press, London, 1984.

De Kleer, J. and J. S. Brown, "A Qualitative Physics Based on Confluences", *Artificial Intelligence,* Vol.24, December 1984, PP7–84.

Dietz, W. E., E. L. Kiech, and M. Ali, "Jet and Rocket Engine Fault Diagnosis in Real Time", *Journal of Neural Network Computing,* Vol.1, No.1, Summer 1989, PP5–18.

Doraiswam, R. and J. Jiang, "Performance Monitoring in Expert Control Systems", *Automatica,* Vol.25, No.6, 1989, PP799–811.

Doukidis, G. I. and R. J. Paul, *Journal of Operational Research Society, special issue on Artificial Intelligence and Expert Systems, Part 1: Production Planning, Application and Methodology,* Vol.41, No.5, 1990.

Doukidis, G. I. and R. J. Paul, *Journal of Operational Research Society, special issue on Artificial Intelligence and Expert Systems, Part 2: Simulation and Scheduling,* Vol.41, No.6, 1990.

Duchessi, P. and R. M. O'Keefe, "A Knowledge-based Approach to Production

Planning", *Journal of Operational Research Society,* Vol.41, No.5, 1990, PP377–390.

Efstathiou, J., "Rule-based Process Control", *Expert Systems and Optimisation in Process Control (1986),* Abe Mamdani and Janet Efstathiou (Ed.), Technical Press, PP43–52.

Efstathiou, J., *Expert Systems in Process Control,* Longman, 1989.

Ellis, J. E., D. S, Wadwani, C. Maxwell, S. N. Das, 1986, "Supervisory Direct Digital Control of A Mixing Process", *The City University Research Memorandum, CEC/JEE-DSW-CM-SND/42.*

Finch, F. E. and M. A. Kramer, "Narrowing Diagnostic Focus Using Functional Decomposition", *AIChE Journal,* Vol.34, No.1, January, 1988, PP25–36.

Forbus, K., "Qualitative Process Theory", *Artificial Intelligence,* Vol.24, December 1984, PP85–168.

Forsyth, R. and Rada, R., *Machine learning: Applications in Expert Systems and Information Retrieval,* Ellis Horwood limited, 1986.

Francis, J. C. and R. R. Leitch, "Artifact: A Real-Time Shell for Intelligent Feedback Control", *Research and Development in Expert Systems (l985),* M. A. Bramer (Ed.), Cambridge University press.

Francis, J. C. and R. R. Leitch, "Intelligent Knowledge-based Process Control", *IEE Control 85,* PP483–488.

Frank, P. M., "Fault Diagnosis on the Base of Dynamic Process Models", *Proc. IMACS 1988,* Paris, July, 1988, Vol.2, PP414–419.

Franks, R. G. E., *Modelling and Simulation in Chemical Process,* Wiley-Interscience, 1972.

Geesey, R. A, and T. A. Blaxton, "Combining Process Models and Knowledge Based Reasoning: The Intelligent Control of Materials Production", *AI Papers,* 1988, the Society for Computer Simulation International, PP12–16.

Haest, M., G. Bastin, M. Gevers, and V. Wertz, "ESPION: an Expert System for System Identification", *Automatica,* Vol.26, No.1, 1990, PP85–95.

Hanakuma, Y., "An Expert System for Fault Diagnosis at Petrochemical Plants", *Future Generation Computer Systems,* Vol.5, No.1, August, 1989, PP103–107.

Harmon, P. and D. King, *Expert Systems,* John Wiley & Sons, Inc. 1985.

Haspel, D. and C. Taunton, "Application of Rule Based Control in the Cement Industry", *Expert Systems and Optimisation in Process Control (1986),* Abe Mamdani and Janet Efstathiou (Ed.), Technical Press.

Herbert, M. R. and G. H. Williams, "An Examination of Qualitative Plant Modelling as a Basis for Knowledge Based Operator Aids in Nuclear Power Stations", *Expert Systems and Optimisation in Process Control (ed Abe Mamdani and Janet Efstathiou),* Technical Press, 1986.

Herbert, M. R. and G. H. Williams, "An Initial Evaluation of the Detection and Diagnosis of Power Plant Faults Using a Deep Knowledge Representation of Physical Behaviour", *Expert Systems,* Vol.4, No.2, May 1987, PP90–99.

Hudlická, E. and V. Lesser, "Modeling and Diagnosis Problemsolving System Behaviour", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol.SMC-17, No.3, May/June, 1987, PP407–419.

Iri, M., K. Aokl, E. O'Shima, and H. Matsuyama, "An Algorithm for Diagnosis of System Failures in the Chemical Process", *Computers and Chemical Engineering,* Vol.3, 1979, PP489–493.

Isermann, R., "Process Fault Detection Based on Modeling and Estimation Methods — A Survey", *Automatica,* Vol.20, No.4, PP387–404, 1984.

Ishida, Y., "An Application of Qualitative Reasoning to Process Diagnosis: Automatic Rule Generation by Qualitative Simulation", *IEEE 4th Conference on Artificial Intelligence Application,* 1988, PP124–129.

Jackson, P., *Introduction to Expert Systems,* Addison-Wesley, 1986.

Johnson, L. and E. T. Keravnou, *Expert Systems Technology: A Guide,* Abacus Press, 1984.

King, R. E. and F. C. Karonis, "Multi-level Expert Control of A Large-Scale Industrial Process", *Fuzzy Computing,* (eds) M. M. Gupta and T. Yamakuwa, Elsevier Science Publisher 1988, PP323–340.

Kramer, M. A., "Malfunction Diagnosis Using Quantitative Models With Non-Boolean Reasoning", *AIChE Journal,* Vol.33, No.1, January, 1987, PP130-140.

Kramer, M. A. and B. L. Palowitch Jr., "A Rule-Based Approach to Fault Diagnosis

Using the Signed Directed Graph", *AIChE Journal,* July 1987, Vol.33, No.7, PP1067–1078.

Kramer, M. A. and B. L. Palowitch Jr., "Expert Systems and Knowledge-based Approaches to Process Malfunction Diagnosis", *AIChE Annual Meeting,* Chicago, IL, 1985.

Kramer, M. A. and F. E. Finch, "Fault Diagnosis of Chemical Process", *Knowledge-based System Diagnosis, Supervision, and Control,* (ed) S, G. Tzafestas, Plenum Press, 1989, PP247–263.

Kuipers, B. J., "Qualitative Simulation", *Artificial Intelligence,* Vol.29, 1986, PP289–338.

Lapointe, J., Marcos, B., Veillette, M., Laflamme, G. and Dumontier, M., "BIOEXPERT — An Expert System for waste Water Treatment Process Diagnosis", *Computers and Chemical Engineering,* Vol.13, No.6, PP619–630, 1989.

Lebow, L. and G. L. Blankenship, "A Microcomputer Based Expert Controller for Industrial Applications", in *Knowledge Based Expert Systems for Engineering: Classification, Education and Control,* (eds) D. Sriram and R. A. Adey, Computational Mechanics Publications, 1987.

Lippmann, R. P., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine,* April 1987, PP4–22.

Lunze, J., "Knowledge-based Systems in Control Engineering", *IFAC/IFORS/IMACS Symposium, Large Scale System 89: Theory and Application,* August 29–31, 1989, Berlin, GDR, P21.

McCluskey, E. G. and S. Thompson, "A Self-Tuning PID Controller Incorporating Expert System Ideas", *IEE Colloquium on "The Use of Expert Systems in Control Engineering",* March 1987, London.

McCluskey, E. G. and S. Thompson, "Rule-based Adaptive PID Control Using a Pattern Recognition Approach", *IEE Colloquium on "Expert Systems in Process Control",* 28 March 1988, London.

Michalski, R. S., J. G. Carbonel and T. M. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach,* Tioga Publishing Company, 1983.

Milne, R., "Strategies for Diagnosis", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-17, NO.3, MAY/JUNE 1987, PP333-339.

Moore, R. L. and Kramer, M. A., "Expert Systems in On-line Process Control", *Chemical Control III,* Asilomer, CA., 1986, PP839–867.

Neat, G. W., H. Kaufman, and R. J. Roy, "Expert Adaptive Control for Drug Delivery Systems", *IEEE Control Systems Magazine,* June 1989, PP20–23.

Nelson, W. R., "REACTOR:, An Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents", *AAAI-82,* August 18–20, 1982, Pittsburgh, Pennsylvania, U.S.A, PP296–301.

Oyeleye, O. O. and M. A. Kramer, "Qualitative Simulation of Chemical Process Systems: Steady-state Analysis", *AIChE Journal,* Vol.34, No.9, September 1988, PP1441–1454.

Pang, G. K. H. and A. G. J. MacFarlane, *An Expert Systems Approach to Computer-Aided Design of Multivariable Systems,* Springer-Verlag, 1987.

Paterson, A., P. Sachs and W. Turner, "ESCORT: the Application of Causal Knowledge to Real-Time Process Control", *Expert Systems 85 (1985),* Martin Marry (Ed.), Cambridge University Press.

Pazzani, M. J., "Refining the Knowledge Base of a Diagnosis Expert System: An Application of Failure-driven Learning", *AAAI-86,* Philadelphia, August 11–15, 1986, PP1029–1035.

Pazzani, M. J., "Failure-driven learning of Fault Diagnosis Heuristics", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol.SMC-17, No.3, May/June, 1987.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann Publishers, 1988.

Petti, T. F., J. Klein, and P. S. Dhurjati, "Diagnostic Model Processor: Using Deep Knowledge for Process Fault Diagnosis", *AIChE Journal,* Vol.36, No.4, April 1990, PP565–575.

Porter, B., A. H. Jones, and C. B. McKeown, "Real-time Expert Tuners for PI Controllers", *IEE Proceedings,* Part D, Vol.134, July, 1987, PP260–263.

Price, C. and M. Lee, "Application of Deep Knowledge", *Artificial Intelligence in Engineering,* Vol.3, No.1, 1988, PP12–17.

Raiman, O., "Order of Magnitude Reasoning", *AAAI-86,* Philadelphia, Aug, 11–15, 1986, PP100–104.

Rao, M., J. P. Tsai, and T. S. Jiang, "An Intelligent Decision Maker for Optimal Control", *Applied Artificial Intelligence,* Vol.2, No.3–4, 1988.

Rasmussen, J., "Models of Mental Strategies in Process Plant Diagnosis", in *Human Detection and Diagnosis of System Failures,* (eds) J. Rasmussen and W. B. Rouse, Plenum, New York, P241, 1980.

Razzak, M. A., T. Hassan, and A. Ahmad, *ExTran User Manual,(Version 7.2),* Intelligent Terminals limited, 1986, Glasgow, U.K.

Rich, S. H. and V. Venkatasubramanian, "Model-Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants", *Computers and Chemical Engineering,* Vol.11, No.2, 1987, PP111–122.

Rich, S. H. and Venkatasubramanian, V. "Causality-Based Failure-Driven Learning in Diagnostic Expert Systems", *AIChE Journal,* Vol.35, No.6, June 1989, PP943–950.

Sanoff, S. P. and P. E. Wellstead, "Expert Identification and Control", *IFAC Identification and System Parameter Estimation 1985,* York, U.K., 1985, PP1273–1277.

Scarl, E. A., J. R. Jamieson, and C. I. Delaune, "Diagnosis and Sensor Validation through Knowledge of Structure and Function", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol.SMC-17, No,3, May/June, 1987, PP360–368.

Shiozaki, J., H. Matsuyama, K. Tano and E. O'shima, "Fault Diagnosis of Chemical Process by the Use of Signed, Directed Graphs, Extension to five-range patterns of Abnormality", *International Chemical Engineering,* Vol.25, No.4, October 1985, PP651–659.

Shum, S. K,, J. F. Davis, W. F. Punch III, and B, Chandrasekaran, "An Expert System Approach to Malfunction Diagnosis in Chemical Plants", *Computers and Chemical Engineering,* Vol.12, No.1, 1988, PP27–36.

Söderström, T. and P. Stoica, *System Identification,* Prentice Hall International, 1989.

Spripada, N. R., D. C. Fisher, and A. J. Morris, "AI Applications for Process Regulation and Servol Control", *IEE Proceedings,* Part D, Vol.134, July, 1987, PP251–259.

Steels, L., "Diagnosis with a Function-Fault Model", *Applied Artificial Intelligence,* Vol.3, No.2–3, 1989, PP213–237.

Tzafestas, S. G. (ed), *Knowledge-based System Diagnosis, Supervision. and Control,* Plenum Press, 1989.

Tzouanas, V. K., W. L. Luyben, C. Georgakis, and L. H. Ungar, "Expert Multivariable Control 1. Structure and Design Methodology", *Industrial and Engineering Chemistry Research,* Vol.29, 1990a, PP382–389.

Tzouanas, V. K., W. L. Luyben, C. Georgakis, and L. H. Ungar, "Expert Multivariable Control 2. Application of EMC to Two-Product Distillation Columns", *Industrial and Engineering Chemistry Research,* Vol.29, 1990b, PP389–403.

Tzouanas, V. K., W. L. Luyben, C. Georgakis, and L. H. Ungar, "Expert Multivariable Control 3. Extension of EMC to Three-Product Side-Stream Distillation Columns", *Industrial and Engineering Chemistry Research,* Vol.29, 1990c, PP404–415.

Vemuri, V. (ed), *Artificial Neural Networks: Theoretical Concepts,* Computer Society of the IEEE, 1988.

Venkatasubramanian, V. and K. Chan, "A Neural Network Methodology for Process Fault Diagnosis", *AIChE Journal,* Vol.35, No.12, December, 1989, PP1993–2002.

Venkatasubramanian, V. and Rich, S, H. "An Object-Oriented Two-Tier Architecture for Integrating Compiled and Deep-level Knowledge for Process Diagnosis", *Computers and Chemical Engineering,* Vol.12, No.9/10, 1988, PP9O3–921.

Wasserman, P. D., *Neural Computing — Theory and Practice,* Van Nostrand Reinhold, New York, 1989.

Watanabe, K., I. Matsuura, M. Abe, N. Kubota, and D. M. Himmelblau, "Incipient Fault Diagnosis of Chemical Process via Artificial Neural Networks", *AIChE Journal,* November 1989, Vol.35, No.11, PP1803–1812.

Waters, A. and J. W. Ponton, "Qualitative Simulation and Fault Diagnosis in Process Plants", *Chemical Engineering Research and Design,* Vol.67, July, 1989, PP407–422.

Yamada, N. and H. Motoda, "A Diagnosis Method of Dynamic System Using the Knowledge on System Description", *IJCAI-83,* Karlsruhe, West Germany, 1983, PP225–229.

Zhang, J., P. D. Roberts, J. E. Ellis, "An Application of Expert Systems Techniques to the On-line Control and Fault Diagnosis of a Mixing Process", *Journal of Intelligent and Robotic Systems,* Vol.1, No.3, 1988, PP209–223.

Bibliography

Andow, P. K., "Fault Diagnosis Using Intelligent Knowledge Based Systems", *Chemical Engineering Research and Design,* Vol.63, November 1985, PP368–372.

Aracil, J., A. Ollero, and A. Garcia-Cerezo, "Stability Indices for the Global Analysis of Expert Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol.19, No.5, September/October 1989, PP998–1007.

Årzén, K. E,, "An Architecture for Expert System Based Feedback Control", *Automatica,* Vol.25, No.6, 1989, PP813–827.

Bandekar, V. R., "Causal Models for Diagnostic Reasoning", *Artificial Intelligence in Engineering,* Vol.4, No.2, 1989, PP79–91.

Bernstein, M., "Learning by Doing — Application of Genetic Search to the Design of Control Strategies", *Proc. IEEE International Symposium on Intelligent Control,* 19–20 January, 1987, Philodelphia, Pennesylvania, PP501–505.

Böhme, B. and V. Starke, "Expert Systems in Process Automation", *IFAC-IFORS-IMACS Symposium, Large Scale System 89: Theory and Application,* August 29–31, 1989, Berlin, GDR, PP326–331.

Bramer, M. A., J. S. Pearce, D. Muirden, J. C. Platts, and D, L. Vipond, "FAUST — An Expert System for Diagnosing Faults in an Electricity Supply System", *Research and Development in Expert Systems V,* (ed) B. Kelly and A. Rector, Cambridge University Press, Cambridge, 1989.

Chandrasekaran, B. and S. Mittal, "Deep verses Compiled Knowledge Approaches to Diagnostic Problem-Solving", *Developments in Expert Systems,* (ed, M. J.Coombs), Academic Press, London, 1984, PP23–34.

Chen, F. C., "Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control", *IEEE Control Systems Magazine,* 1990, PP44–48.

Chung, D. T., M. Modarres, and R. N. M. Itunt, "GOTRES: An Expert System for Fault Detection and Analysis", *Reliability Engineering and System Safety,* Vol.24, 1989, PP113–137.

Cross, S. E., "An Application of Qualitative Simulation to Expert System Plant Justification", *SIGART Newsletter,* July 1985, No.93, PP9–12.

Daniel, S. W., "Comparative Analysis", *AAAI-86,* Philadelphia, Aug. 11–15, 1986, PP959–965.

Darnall, S. P., "A Knowledge-Based Simulation for the Development of A Real-Time Avionic Expert System", *AI Papers,* 1988, the Society for Computer Simulation International, PP32–35.

De Kleer, J. and B. C. Williams, "Reasoning about Multiple Faults", *AAAI-86,* Philadelphia, Aug.11–15, 1986, PP132–139.

Forbus, K., "Qualitative Physics", *SIGART Newsletter,* July 1985, No.93, PP7–9.

Fortuna, L., A. Gallo, and G. Nunnari, "Expert System for Optimization of Model Reduction Techniques", *IEEE Control Systems Magazine,* June 1989, PP9–13.

Frank, P. M., "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy — A Survey and Some New Results", *Automatica,* Vol.26, No.3, PP459–474, 1990.

Freeman, D. D., "Artificial Intelligence Applications in Process Control", *Proc. ACC,* 1985, PP889–891.

Garrison, D. B., D. M. Prett, and P. E. Steacy, "Expert Systems in Process Control and Optimisation — A Perspective", *Shell Workshop on Process Control,* (eds, D. M. Prett and C. M. Garcia), 1987, PP37–48.

Goldberg, D. E., "Dynamic System COntrol Using Rule Learning and Genetic Algorithms", *Proceedings of the 9th International Joint Conference on Artificial Intelligence,* Vol.1, PP588–592.

Grantham, S. D. and L. H. Ungar, "A First Principles Approach to Automated Troubleshooting of Chemical Plants", *Computers and Chemical Engineering,* Vol.14, No.7, 1990, PP783–798.

Greiner, R., B. A. Smith, and R. W. Wilkerson, "A Correction to the Algorithm in Reiter's Theory of Diagnosis", *Artificial Intelligence,* Vol.41, No.1, November 1989, PP79–88.

Gupta, U. K. and M. Ali, "Hierarchical Representation and Machine Learning from Faulty Jet Engine Behavioural Examples to Detect Real Time Abnormal Conditions", *The First International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Tennessee, June 1–3, 1988, PP710–720.

Hudlicka, E. and V. R. Lesser, "Construction and Used of a Causal Model of a

Problem-Solving System", *SIGART Newsletter,* July 1985, No.93, PP48-54.

Ioneseu, D. and I. Trif, "Hierarchical Expert System for Process Control", *Proc. ACC,* Vol.1, 198B, PP1-6.

Karsai, G., C. Biegl, S. Padalkar, J. Sztipanovits, K. Kawamura, N. Miyasuka, and M. Inui, "Knowledge-based Approach to Real-time Supervisory Control", *Proc. ACC,* 1988, PP620-625.

Ke, M. and M. Ali, "MLS, A Machine Learning System for Engine Fault Diagnosis", *The First International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Tennessee, June 1-3, 1988, PP721-727.

Kodratoff, Y. and A. Hutchinson (eds), *Machine and Human Learning,* Michael Horwood Limited, 1989.

Koukoulis, C., "A frame-based Method for fault Diagnosis", *Expert Systems and Optimisation in Process Control (1986),* Abe Mamdani and Janet Efstathiou (Ed.), Technical Press, PP160-174.

Krijgsman, A. J., H. B. Verbruggen, and P. M. Bruijn, "Knowledge-based Real-time Control", *IFAC Artificial Intelligence in Real-time Control,* Swansea, U. K., 1988, PP13-19.

Kristinsson, K. and G. A. Dumont, "Genetic Algorithms in System Identification", *Proceedings of IEEE International Symposium on Intelligent Control 1988,* 24-26 August, 1988, Arlington, Virginia, U.S.A., PP597-602.

Leary, J. J. and P. J. Gawthrop, "Process Fault Detection Using Constraint Suspension", *IEE Proceedings,* Part D, Vol.134, July, 1987, PP264-271.

Lee, S. and M. H. Kim, "Cognitive Control of Dynamic Systems", *Proc. IEEE International Symposium on Intelligent Control,* 19-20 January, 1987, Philodelphia, Pennesylvania, PP455-460.

Li, W. and J. E. Slotine, "Neural Network Control of Unknown Nonlinear Systems", *Proc. ACC,* 1989, PP1136-1141.

Liu, K. and J. Gertler, "Monitoring the Condition of Feedback Control Systems", *Proceedings of IEEE International Symposium on Intelligent Control 1988,* 24-26 August, 1988, Arlington, Virginia, U.S.A., PP51-56.

Lu, Y. Z., "A Marriage Between Modern System Control and Artificial Intelligence — the Future of Industrial Control", *Proceedings of MICONEX 88,* PP29–43.

Luger, G. F. and W. A. Stubblefield, *Artificial Intelligence and the Design of Expert Systems,* The Benjaming/Cummings Publishing Company, Inc. 1989.

McGhee, J., M. J. Grimble, and P. Mowforth (eds), *Knowledge-based Systems for Industrial Control,* Peter Peregrinum Ltd, 1990.

Mahmoud, M. S., S. I. Eid, and A. A. Abou-Elsoud, "A Real-Time Expert Control System for Dynamical Processes", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol.19, No.5, September/October 1989, PP1101–1105.

Abe Mamdani and Janet Efstathiou (Ed.), *Expert Systems and Optimisation in process Control,* 1986, Technical Press.

Matwin, S., F. Oppacher, and P. Constant, "Knowledge Acquisition by Incremental Learning form Problem — Solution Pairs", *Computational Intelligence,* Vol.5, No.2, May 1989.

Naidu, S., E. Zafiriou, and T. J. McAvoy, "Application of Neural Networks on the Detection of Sensor Failure During the Operation of a Control System", *Proc. ACC Meeting,* Pittsburgh, 1989, PP1336–1341.

Narendra, K. S. and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Transactions on Neural Networks,* Vol.1, No.1, March 1990.

Nguyen, D. and B. Widrow, "The Truck Backer-Upper: An Example of Self-learning in Neural Networks", *International Joint Conference on Neural Networks,* 1989, Vol.2, PP357–363.

Niida, K., J. Itoh, T. Umeda, S. Kobayashi, and A. Ichikawa, "Some Expert System Experiments in Process Engineering", *Chemical Engineering Research and Design,* Vol.64, September 1986, PP372–380.

Niwa, Y., T. Sumigama, M. Sakurai, and T. Aoki, "Application of a Self-learning Function to an Expert System for Blast Furnace Heat Control", *ISIJ International,* Vol.30, No.2, 1990, PP111–117.

O'Neill, D. M. and P. W. Mullarkey, "A Knowledge-based Approach to Real-time Signal Monitoring", *IEEE 5th Conference on Artificial Intelligence Applications,* Miami, Florida, March 6–10, 1989, PP134–141.

Palowitch, B. L. and M. A. Kramer, "The Application of a Knowledge-based Expert

System to Chemical Plant Fault Diagnosis", *Proc. ACC,* 1985, PP646-651.

Pazzani, M. J., "Explanation-based Learning of Diagnostic Heuritics: A Comparison of Learning from Success and Failure", *Proc. 1989 Annual Artificial Intelligence Systems in Government Conference,* March 27–31, 1989, Washinton, D. C., PP164–168.

Price, C. and J. Hunt, "Using Qualitative Reasoning to Build Diagnostic Expert Systems", *Research and Development in Expert Systems VI,* (ed) N. Shadbolt, Cambridge University Press, 1989.

Ramamoorthy, P. A. and S. Huang, "Implementation of Rule-Based Expert Systems for Time-critical Applications Using Neural Networks", *IEEE International Conference on System Engineering,* U.S.A., August 24–26, 1989, PP147–150.

Ramesh, T. S., S. R. Shum, and J. F, Davis, "A Structured Framework for Efficient Problem Solving in Diagnostic Expert Systems", *Computers and Chemical Engineering,* Vol.12, No.9–10, 1988, PP891–902.

Reiter, R., "A Theory of Diagnosis from First Principles", *Artificial Intelligence,* Vol.32, 1987, PP57–96.

Rhee, F. V. D., H. R. V. N. Lemke, and J. G. Dijkman, "Knowledge Based Fuzzy Control of Systems", *IEEE Transactions on Automatic Control,* Vol.35, No.2, February 1990, PP148–155.

Rodd, M. G., H. T. Li, and S. Q. Su, *Proceedings of IFAC Workshop on Artificial Intelligence in Real-time Control,* Shenyang, China, 19–21 September, 1989.

Saridis, G. N., "Knowledge Implementation: Structures of Intelligent Control Systems", *Proc. IEEE International Symposium on Intelligent Control,* 1987, Philadelphia, PP9–17.

Shadbolt, N. (ed), *Research and Development in Expert Systems VI,* Cambridge University Press, 1989.

Shirley, R. S., "Some Lessons Learned Using Expert Systems for Process Control", *IEEE Control Systems Magazine,* December, 1987, PP11–15.

Song, Z., J. C. Gao, and C. H. Zhou, "The Rule-based Distillation Decision and Control System", *Proceedings of IFAC Workshop on Artificial Intelli-*

gence in Real-time Control, Shenyang, China, 19–21 September, 1989, PP149–154.

Steels, L. and W. Van de Velde, "Learning in Second Generation Expert Systems", *Machine and Human Learning,* (eds, Y. Kodratoff and A. Hutchinson), Michael Horwood Limited, 1989, PP53–75.

Stephanon, H. E., A. Meystel, and J. Y. S. Luh (eds), *Proceedings of IEEE International Symposium on Intelligent Control 1988,* 24–26 August, 1988, Arlington, Virginia, U.S.A.

Subramanian, C. and M. Ali, "Detecting Abnormal Situations from Real Time Power Plant Data Using Machine Learning", *The First International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Tennessee, June 1–3, 1988, PP738–746.

Sullivan, G. and L. K. Lauderbungh, "Expert Aided Adaptive Control", *Proceedings of IEEE International Symposium on Intelligent Control 1988,* 24–26 August, 1988, Arlington, Virginia, U.S.A., PP574–579.

Sztipanovits, J., G. Karsai, and L. Biegl, "Modelling, Model Interpretation and Intelligent Control", *Proceedings of IEEE International Symposium on Intelligent Control 1988,* 24–26 August, 1988, Arlington, Virginia, U.S.A., PP46–50.

Taylor, J. H., R. W. Gerbarolt, and E. C. Luce, "Rule-Based Real-Time Control Systems", *Proceedings of IEEE Decision and Control,* 1987, Vol.3, PP 1923–1928.

Tinham, B., "Knowledge Maketh the Control System", *Control and Instrumentation,* February, 1990, PP57–60.

Torasso and L. Console, *Diagnostic Problem Solving,* North Oxford Academic, 1989.

Tzafestas, S. G., S. Abu El Ata-Doss, and G. Papakonstantinou, "Expert System Methodology in Process Supervision and Control", *Knowledge-based System Diagnosis, Supervision, and Control,* (ed) S. G, Tzafestas, Plenum Press, 1989, PP181–215.

Verbruggen, V. B. and K. J. Åström, "Artificial Intelligence and Feedback Control", *Proceedings of IFAC Workshop on Artificial Intelligence in Real-time Control,* Shenyang, China, 19–21 September, 1989, PP1–11.

Wang, S. B. and Y. Z. Lu, "Knowledge-based Successive Linear Programming for Linearly Constrained Optimization", *International Journal of Man-Machine Studies,* Vol.29, 1988, PP625–636.

Waye, D., A. Terroax, and S. Goyette, "SIMSMART for Process and Process Control Optimisation: An Application Using the Synthetic Intelligence Approach", *AI Papers,* 1988, the Society for Computer Simulation International, PP91–96.

Xu, H. Y., C. R. Baird, and D. Riordan, "An Intelligent Tuning Control Technique", *Applied Artificial Intelligence,* Vol.4, 1990, PP95–108.

Zadeh, L., "Knowledge Evaluation: Observation and Computation for Intelligent Control", *Proc. IEEE International Symposium on Intelligent Control,* 1987, Philadelphia, PP302–308.

Zhou, H. H., "A Study of Automatic Rule Generation and Adaptation", *The First International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Tennessee, June 1–3, 1988, PP747–752.

Zhu, Z, and G. Dai, "Expert System Approach of Industrial Process Adaptive Control", *Proc. IFAC Workshop on Robust Adaptive Control,* Newcastle, Australia, 22–24, Aug.1988, PP57–61.