FAULT DETECTION AND IDENTIFICATION IN A DEEP TROUGH HYDROPONIC SYSTEM USING ADAPTIVE NEURO-FUZZY ANALYSIS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Albert Setiawan

May 2008

FAULT DETECTION AND IDENTIFICATION IN A DEEP TROUGH HYDROPONIC PLANT PRODUCTION SYSTEM USING ADAPTIVE NEURO-FUZZY ANALYSIS

Albert Setiawan, Ph. D.

Cornell University 2008

An early fault detection and identification system (FDI) can be an important part in any plant production system. A FDI can be used to avoid costly repairs and long disruptions in production. A hydroponic plant production system is a complex biological system that contains plants and microorganisms in its processes that are hard to model mathematically. A soft computing method called a neuro-fuzzy system is chosen to implement the FDI. A neuro-fuzzy system is a hybrid combination of a neural network and a fuzzy logic system that combines the best from both methods: knowledge based structure from fuzzy logic and a proven learning capability from a neural network. An adaptive neuro-fuzzy inference system (ANFIS) is developed to detect and identify actuator and sensor faults in the hydroponic plant production system. A separate system for exploring the ANFIS capability in detecting biological faults is also investigated. The novelty of the neuro-fuzzy FDI in this research used a single output to simultaneously detect and identify various faults in the system.

BIOGRAPHICAL SKETCH

Albert Setiawan was born in Jakarta, Indonesia, 1969. He graduated from Electrical Engineering Department of University of Indonesia in 1992 with specialization in Control and Instrumentation. He received his Master of Science Degree from Agricultural and Biological Engineering of Cornell University in 1998 specializing in Controlled Environment Agriculture (CEA).

ACKNOWLEDGEMENTS

I am especially grateful to my advisor Prof. Louis D. Albright for his patience in guiding and teaching me through my years in Cornell. His help and support made this dissertation possible. I want to thank my minor committee members, Prof. Robert W. Langhans and Prof. Daniel Aneshansley for their support and advice during my graduate study. I also want to thank Konstantinos Ferentinos (Dinos) for his friendship, help and discussion. Jennifer Mathieu's help in maintaining and improving the hydroponics system during experiments is significant. I am grateful to Andy Leed and the KPL greenhouse staffs that are very knowledgeable and helpful during my research in Cornell. I also want to thank my parent for their selfless support, my sister Angela for providing me place to stay and finish my PhD. Thank you also to Mallory Parmerlee who, despite his busy schedule, helped me with editing the manuscript. Last but not least I want to thank my wife, Lina, who stands by me no matter what and my daughters, Elysia and Vanya who always cheer me up.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Hydroponic plant production systems are known for their high quality products. A reliable and precise environment control system is critical to achieve this goal. The environmental control system monitors and controls the nutrient solution variables (pH, electrical conductivity, dissolved oxygen and temperature) and greenhouse aerial environment accurately.

Today's digital controls and computers are becoming more common in automating greenhouse operations, replacing many independent analog sensors and controllers that frequently work against each other and inflate the operational cost. A grower can set a detailed plant production schedule in the computer according to consumer demand. The schedule can be executed efficiently and timely. The result is better quality plants with less cost.

As the plant quality and harvest timing become important issues in maximizing profit according to season, any fault in the system can delay scheduled production or even destroy valuable crops. A fast responding fault detection scheme as a part of the hydroponics system is crucial to guarantee continuous and optimal production. Since many growers already use computers in their greenhouses, applying the fault detection scheme adds minimal cost to the grower's system.

There are two ways to detect faults in the system: by redundancy and by interaction with other variables. Redundancy fault detection uses redundant components to detect malfunctions or faults. For example, a system with an additional sensor can tell whether one or both sensors are malfunctioning when their outputs are significantly different from each other. The other fault detection scheme detects faults indirectly. Since most variables interact with each other and influence one another, a malfunction can be detected by unusual interaction with other variables. Some detection schemes combine these two techniques since some faults are better detected with redundancy and some can be detected reliably by indirect interaction with other variables. In this thesis, the indirect way of fault detection will be explored. Since the indirect way of detection does not require additional hardware, its application will be interesting for growers who want additional insurance for their crops with minimal additional cost.

The environment inside the greenhouse is subjected to many disturbances. Outside conditions, such as wind speed and direction, humidity, sunlight, clouds, rain and snow vary diurnally, seasonally and sometimes randomly. In addition to these factors, the hydroponic system itself is a complex nonlinear system involving biological processes. Interactions between plants, nutrient solution and the microorganism population affect the solution variables and add complexity. This is hard to quantify. In fact, most real word applications involve uncertainties which might vary randomly and cannot be predicted a priori. Fuzzy inference systems have been developed to deal with this issue. In particular, a neuro-fuzzy system is a good candidate for fault detection and identification systems since it combines the best of fuzzy and neural network. It has both a structured knowledge base of fuzzy logic and a learning paradigm from the artificial neural network. Neuro-fuzzy fault detection and identification schemes will be explored in this dissertation in the context of a hydroponic plant production system.

CHAPTER 2

BACKGROUND ON FAULT DETECTION

The majority of fault detection and identification (FDI) schemes consist of residual generation and residual analysis (Koppen-Seliger and Frank, 1999) or residual generation and decision making (Bocaniala and Palade, 2006).

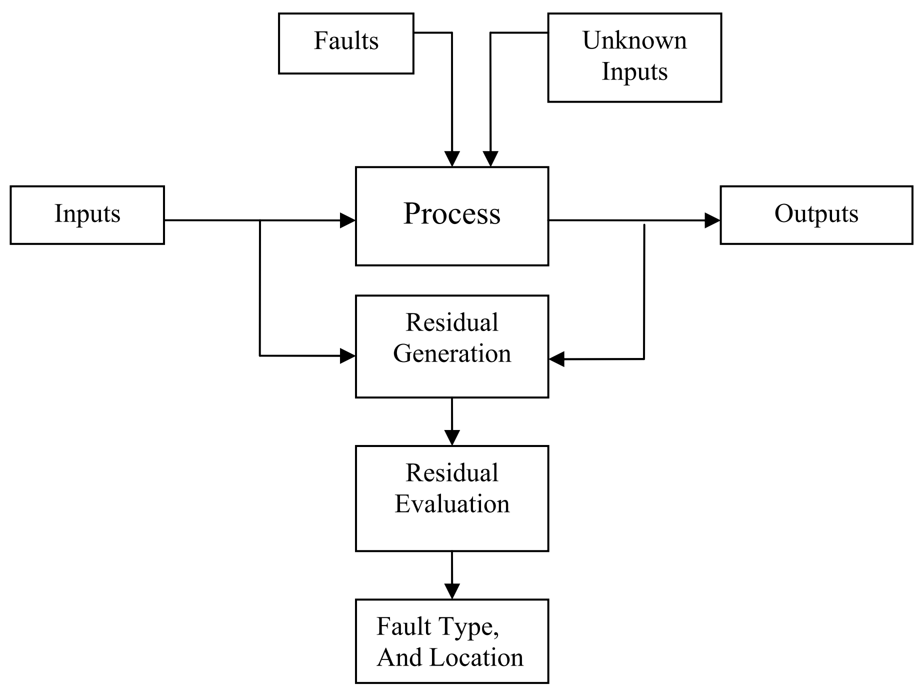

Figure 2.1 Fault Detection and Identification (FDI) Scheme with Residual Generation (Koppen-Seliger and Frank, 1999)

The diagram of this FDI scheme can be seen in Figure 2.1. Signals called residuals are generated in the residual generation stage. Residuals are the inconsistencies between the data from the system measurements and the corresponding signals of the model (Mendonca et al., 2006). These residuals are the fault indicators that reflect the faulty condition of the monitored system. A residual generation is followed by a residual evaluation. In this stage a monitored system condition is evaluated for a fault detection and identification. The outputs of this stage are time of occurrence, fault type and location.

The residual generation stage is usually based on analytical or mathematical models. This includes linear and non-linear models. Sometimes it is difficult to obtain accurate mathematical models as in the case of complex systems.

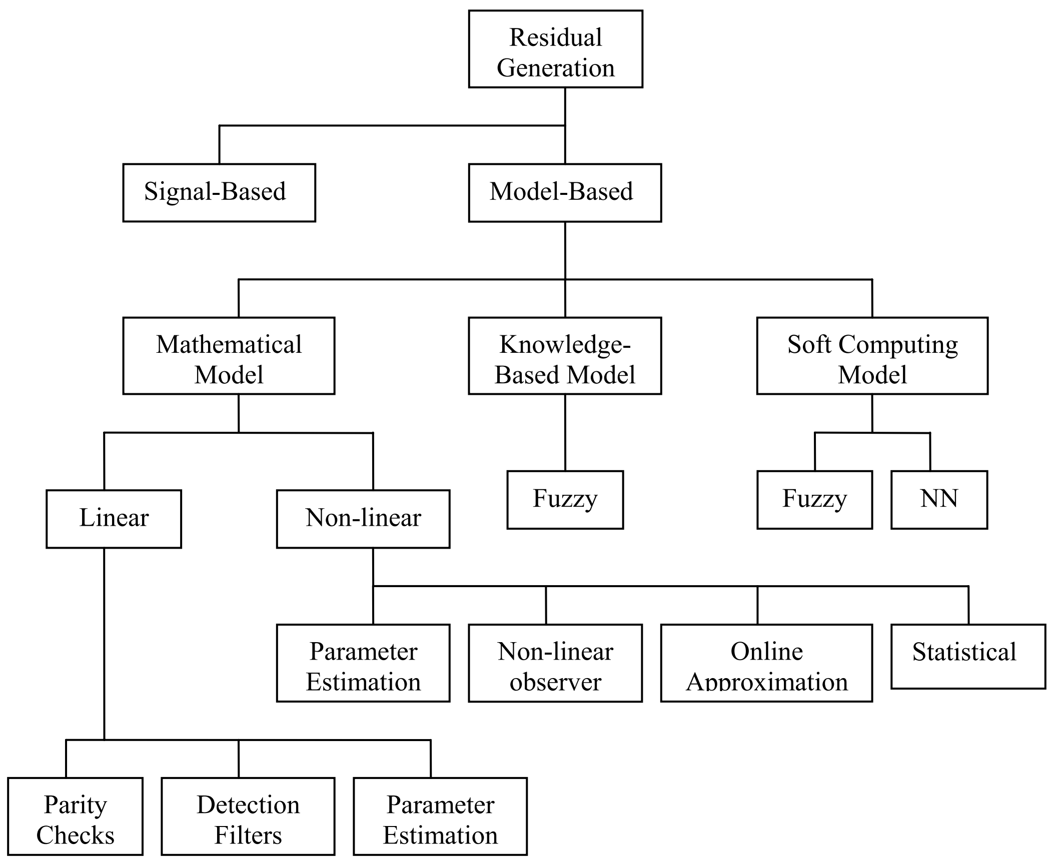

Figure 2.2 The Residual Generation Methods Modified From
Koppen-Seliger and Frank (1999)

Fuzzy systems, neural networks and other new emerging techniques known as soft computing have been developed in recent years to solve this problem (Calado et al, 2001). A diagram of various methods of a residual generation inspired by the one from Koppen-Seliger and Frank (1999) are shown in Figure 2.2.

A residual evaluation can be as simple as a threshold decision or it can use statistical and pattern recognition methods. Different residual evaluation methods can be seen in Figure 2.3. Classification techniques such as the fuzzy logic and the neural network are natural tools in detecting and identifying faults in residuals. Recently these methods have gained popularity as residual evaluation methods (Calado et al, 2001).

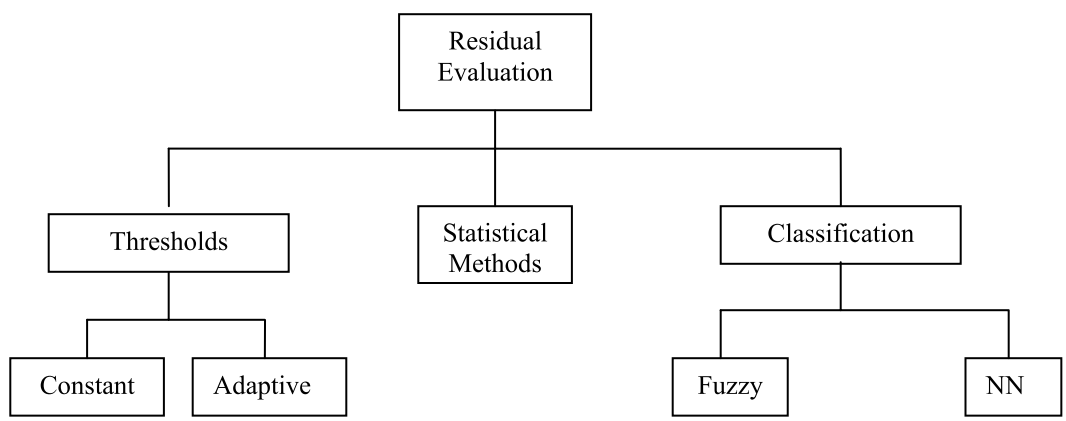

Figure 2.3 Residual Evaluation Methods Modified from Koppen-Seliger and Frank (1999)

The trend these days is to combine different methods to develop a hybrid fault detection system. An example for such a hybrid is the use of a mathematical model for the residual generation and the neuro-fuzzy for the residual analysis. Several examples of this hybrid are described below.

A neuro-fuzzy system was first applied to the fault diagnosis of an automotive electromechanical actuator (Pfeufer, 1997). The electromechanical actuator is used for automotive applications such as traction and velocity control. This fault diagnosis approach has two mathematical models involving seven different parameters of the actuator such as armature resistance, magnetic flux linkage, moment of inertias, viscous friction coefficient, spring constants etc. The system's output was compared to the normal values of the fault free case from the models and the deviations (residual) of the parameters were considered as the fault symptoms. These deviations were used as inputs for 14 independent neuro-fuzzy systems, each of which was sensitive for one kind of fault in the system. The neuro-fuzzy systems for the fault diagnosis had 18 to 28 rules. These rules were formed from the training data set by a rule extraction algorithm. The fault diagnosis system was able to classify 98.5% of the faults. The misclassification was caused by high disturbances on the related symptoms relative to the changes of the mean values and the lack of differences between the symptom patterns.

A hybrid artificial neural network with fuzzy rule based decision making of sensor fault detection, isolation and accommodation in automotive engines was proposed (Capriglione et al, 2003). The fault detection system used two independent neural networks, each with a different combination of inputs for generating throttle output. The inputs included the previous 3 to 5 steps of data, which was needed for small fault detection. The throttle outputs of the neural network models were compared to the actual data to generate two residuals. If one or both of the residuals were outside of the determined threshold values, a fault was present in the system. Heuristic fuzzy rules then were used to identify which sensor was faulty based on the pattern of the residual values of throttle sensor, manifold pressure sensor or crankshaft speed sensor. After the identity of the faulty sensor was found by this method, another neural network model was used to classify the type of sensor fault. The sensor faults were classified as open circuit, short circuit, hold, short circuit between two sensors and miscalibration. The scheme was able to detect 100% of the faulty conditions and about 90% of correct isolation/identification.

Since neuro-fuzzy systems (NFS) and artificial neural networks (ANN) are used for both residual generation and residual analysis, it is logical to develop just one system for the fault detection and identification directly from input-output data in order to reduce modeling errors and computation time of the two different models. Some researchers have attempted this method with the ANN (Sorsa, 1991, Ferentinos, 2002).

Sorsa (1991) compared three different ANN to develop a fault detection and diagnosis on a simulated heat exchanger-continuous stirred tank reactor system: a single layer perceptron (SLP), a multilayer perceptron (MLP) and a counterpropagation network. The models had 14 inputs and 10 different faults as the outputs. Simulated noise was added to the measurements that varied from 0% to 10% of the measurement region. The representative faults in the system were: 1) Input pipe partially blocked, 2) Recycle pipe partially blocked, 3) Input concentration of A high, 4) Recycle flow set point high, 5) Fouled Heat Exchanger, 6) Deactivated Catalyst, 7) Temperature control valve stuck high, 8) Leak flow in reactor, 9) Recycle flow meter stuck high, and 10) Malfunction in pump.

The SLP has 14 input nodes and 10 output nodes. Each output is used to examine one faulty condition in the monitored system. The output nodes use a sigmoid activation function. The normal condition should produce all outputs near zero. A particular fault produced an output value of one in the corresponding output and zero in the other outputs. The three different ANNs were trained 5,000 times.

The MLP has 14 input, 4 hidden and 10 output nodes. A sigmoid activation function was used for both hidden and output nodes. This configuration gave a better fault detection than SLP. Changing the hidden nodes activation function from sigmoidal to hyperbolic tangents drastically reduced the training time. The addition of a second hidden layer significantly added to the computation time and reduced generalization.

The counterpropagation network in Sorsa (1991) has a Kohonen layer and a Grossberg layer. More components in the Kohonen layer increase successful classifications. The best counterpropagation network still failed to classify fault 2 and fault 10. The MLP gave the best result from all three different NNs. This paper shows that a direct input-output fault detection and identification system can be successfully formed for a complex system (14 inputs and 10 outputs). Although the method used was a neural network, a comparable neuro-fuzzy fault detection and identification system can be developed as well.

Ferentinos (2002) used MLP to detect and identify faults in a deep-trough hydroponics system. He tried several hidden layers and concluded that a single hidden layer performed the best. A genetic algorithm was used to choose the best NN architecture, including the activation function and learning method. The comparison of NN application in fault detection and identification in Ferentinos' work with neuro-fuzzy method can be seen in chapter 8.

A fuzzy or neuro-fuzzy system as a single system has not been explored as well as NN for detecting and identifying faults in a complex system. A neuro-fuzzy system is especially promising since it combines the advantages of both neural network and fuzzy logic. The resulting system has a clear knowledge base in the form of IF THEN rules and should perform as well as a neural network.

Shukri (2004) developed a simple adaptive neuro-fuzzy inference system (ANFIS) model with 2 inputs and one output to detect the condition of an induction motor. The model estimated the friction which was developed in the motor over time that was caused by a bearing failure. The output of the neuro-fuzzy system was three singletons to represent the condition as good, fair and bad. Although the result was very encouraging and was able to correctly identify the condition, the whole experiment was done without real world data and with only a few inputs. The system was based on simulation data which was generated from an asynchronous motor model found in MATLAB's SIMULINK library.

It was shown above in Pfeufer (1997) that a quite complex neuro-fuzzy fault detection and identification system with seven inputs can be built based on residuals. It is generated from the discrepancy between the process measurements and the corresponding signals of the mathematical model that can be considered as 'filtered data'.

This dissertation extends this limit by using real world input-output data to directly develop the neuro-fuzzy fault detection and identification systems with as many as 39 inputs and only a single output to simultaneously detect and identify various faults in the system. This is accomplished by carefully choosing the inputs and the neuro-fuzzy system with the most effective pattern recognition. A Neuro-fuzzy system based on a radial basis function in constructing Takagi-Sugeno (TS) rules (Takagi, 1985) was chosen for this task based on its capability as an efficient universal approximator.

CHAPTER 3

OBJECTIVES

This dissertation attempts to develop a fault detection and identification system for deep trough hydroponics plant production using a neuro-fuzzy algorithm.

The specific objectives of this study are:

1. To derive a neuro-fuzzy fault detection and identification system that is easy to use for hydroponic plant production systems using environmental parameters of the hydroponics system.
2. To optimize the neuro-fuzzy fault detection and identification system for hydroponic plant production systems.
3. To compare the results with a multi layer perceptron neural network fault detection and identification system developed for the same system.

CHAPTER 4

EXPERIMENTAL SETUP AND METHOD

4.1 Deep-Trough Experimental Setup

4.1.1 The Greenhouse Section

The experiments were conducted in section D of greenhouse #15 in Kenneth Post Laboratory, Cornell University, Ithaca, NY. This greenhouse had 5 identical sections (A-E). Each section had a floor area of 85 m^2. A central computer controlled the aerial environmental parameter of every greenhouse section via Analog Device's 6B microcontroller module (details in Appendix A).

The temperature set points were 19C during the night and 24C during the day and were mostly achieved within ± 0.5C. The greenhouse also had staged ventilation, evaporative cooling, and a movable shading system for cooling control.

The light intensity was measured using a LI-COR quantum sensor that gave the readings of light intensity in 400-700 nm wavelengths needed for plant photosynthesis. The daily photosynthetically active radiation (PAR) integral set point was 17 $mols/m^2$. This was achieved by using supplemental lighting from twenty-one high-pressure sodium (HPS) 400 W lamps that gave uniform light intensity of 200 $\mu molm^{-2}s^{-1}$ at the top of plant canopy.

Relative humidity and CO_2 were also continuously monitored. The relative humidity was maintained between 30% and 70%. The central computer sent the control signal and logged the data every two minutes.

4.1.2 The Cultivation System

The deep trough hydroponic system consisted of 3 small growing ponds (stainless steel tanks) with a dimensions of 121cm x 60cm x 28cm. The tanks were filled with nutrient solution to a certain level and the plants were placed in floating styrofoam panels. The nutrient solution surface was completely covered with styrofoam panels to reduce evaporation and discourage algae growth. One of the tanks was used as a control and the other two were used for fault treatments.

Lettuce (*Lactuva Sativa cv Vivaldi*) seeds were placed into a hole in the center of small rock wool cubes filled with peatlite to facilitate uniform germination. From day one to day eleven the seedlings were grown in a growth chamber. The environmental setting was similar to the greenhouse except for the chamber's 24-hour lighting period. On day twelve, the seedlings were transplanted to the experimental tanks occupying two rows. Each row consisted of 3 and 4 plants placed in alternating fashion. Styrofoam spacers of 2 cm thickness were inserted to give additional spacing for the plants so the leaves would not overlap with the neighboring plants, which occurred after twenty days. The next older generation had an additional spacer between them. The 27 day old plants were harvested every two days to make room for the new generation of plants. The layout of plant placement in the tank can be seen in Figure 4.1. A continuous plant production system was developed with this arrangement so the result would be directly applicable to the commercial hydroponic plant production system.

The nutrient solution was circulated through a filter and dispersed uniformly through small holes in the pipes along the perimeter of the system. The pipes were also used for acid/base injection to maintain pH so that damage to the roots from any direct contact with pure acid could be avoided. Pure oxygen was also injected into the circulation system to maintain the optimal oxygen level since the nutrient solution surface was completely covered. Fresh nutrient concentration and the water level in each tank were maintained every two days to assure that the nutrition solution remained at the desired level.

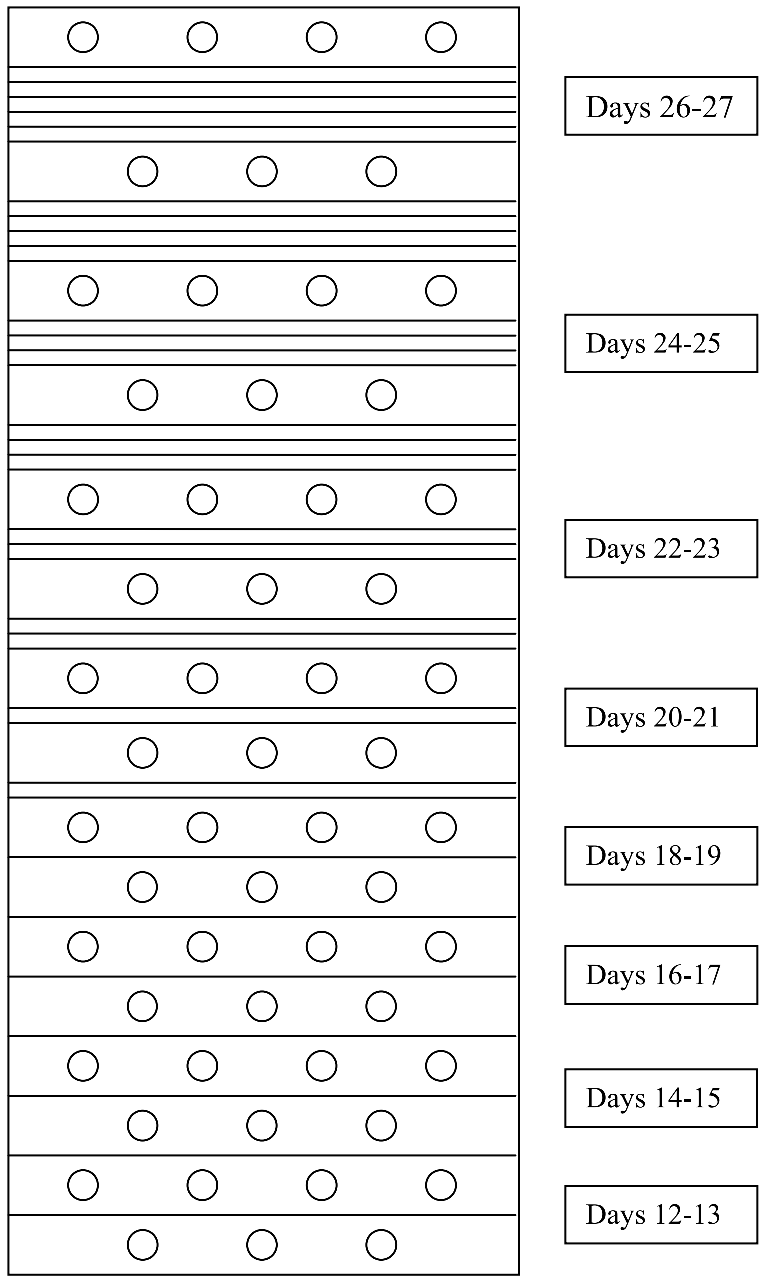

Figure 4.1 Plant Spacing in The Hydroponic System

LabView from National Instruments was used to control and monitor variables in the nutrient solution such as temperature, electrical conductivity (EC), pH, dissolved oxygen (DO), nitrate concentration and transpiration. Sensors were connected to their corresponding meters and their outputs were connected to a data acquisition system from National Instruments. The sensor assembly inside the tank can be seen in Figure 4.2. This computer dealt with the environmental parameters of the root zone of the hydroponic system while the central computer dealt with the aerial environmental parameters of the greenhouse section. The program controlled and monitored the nutrient solution of the three tanks independently every 10 seconds and logged the data every 5 minutes. The pseudo-derivative feedback (PDF) control algorithm was used (details in Setiawan, 1998) which is good at dealing with the external disturbance. The detailed connection schematic of the sensors and equipments can be found in Appendix A.

Figure 4.2 Sensor Assembly in The Tank

From top to bottom: pH, DO, EC.

The pH was maintained at 5.8 using a metering pump, which injected additional acid (1M HNO_3) needed for pH control. The DO was maintained between 6.5mg/l and 7 mg/l by controlling the flow of oxygen from a tank using a solenoid valve. The EC was maintained manually between 1150 to 1250 μS/cm. A scale was used to weigh the whole tank to calculate the transpiration rate. Nitrate concentration in the nutrient solution was an important variable to be monitored since the nitrate uptake was a good indicator of plant growth and thus a good indicator of plant stress. After considerable searching, a reliable and robust nitrate analyzer could not be found and nitrate concentration was not used for fault modeling.

4.2 Methods

A fault detection in the hydroponic system can be divided into two groups: sensor/actuator or mechanical faults and biological faults. This division is needed since they have different time constants and use different inputs. Transpiration rate, which is the main variable for any biological fault detection system, was not used for the mechanical fault detection system.

Mechanical faults can be divided into abrupt faults and incipient faults. Four kinds of mechanical faults were imposed into the hydroponic plant production system. Failure of the pH control pump and the circulation pump represented abrupt faults. Drifting of the pH sensor and EC sensor represented incipient faults. The data from several repetitions of fault experiments were used to develop the neuro-fuzzy fault detection systems.

Biological faults are imposed directly on the plants. These faults can be divided into shoot and root faults. There were four different series of experiments to mimic the effect of possible faults in the plants. The first one was to remove the largest plants (of ages of 25 and 27 days) from the tanks and allow the roots to be exposed to air for five minutes. This treatment caused a slight disturbance to the roots of the plants. The second experiment involved bruising the leaves of the largest plants (of ages of 25 and 27 days). This treatment simulated a fault occurring in the shoot of the plants. The third plant disturbance was to remove most of the leaves from three generations of the largest plants (23, 25 and 27 days old). This represented a major fault in the shoot zone. The last experiment was to cover the leaves of the largest plants (23, 25 and 27 days old) with plastic bags to simulate a major problem in the root zone. This last treatment drastically reduced the transpiration rate. As in the mechanical fault, several neuro-fuzzy fault detection systems were developed and compared.

CHAPTER 5

FAULT DETECTION AND IDENTIFICATION MODEL DEVELOPMENT

Analytical model-based techniques represent the majority of fault detection and isolation methods in the literature (Simani et al, 2003). The statistics show that the number of applications using nonlinear mathematical models is growing while the trend of using linear mathematical models is diminishing. However, it is difficult to achieve accurate nonlinear mathematical models for complex nonlinear systems. If the system structure is not completely known, the fault diagnosis should be based on data or heuristic information. The inherent characteristics of fuzzy logic are suitable for fault detection and isolation of complex nonlinear systems. The nonlinear mapping characteristic of a fuzzy model, with fast and robust implementation, and the capacity to embed a priori knowledge and the ability of generalization can be beneficial to fault detection (Mendonca et al, 2006). With these advantages, a fuzzy model is a natural tool to deal with nonlinear and uncertain conditions in the hydroponic plant production system.

5.1 Fuzzy Logic

The core of fuzzy logic is the fuzzy set (Zadeh, 1965) and the IF THEN knowledge base (Zadeh, 1973). The fuzzy set is a set without a crisp boundary. There is a gradual transition between something that belongs and something that doesn't belong to a set. This is characterized by a membership function with values between zero and one. Zero means it definitely does not belong to the set and one means it definitely does belong to the set. The number between these two limits represents the degree of membership in that set. A membership function is usually symbolized by μ. For example, the normal greenhouse temperature during the day is about 24 C. In this case the temperature value comes from a sensor reading as a crisp value and this number should be transformed (fuzzified) into a fuzzy number.

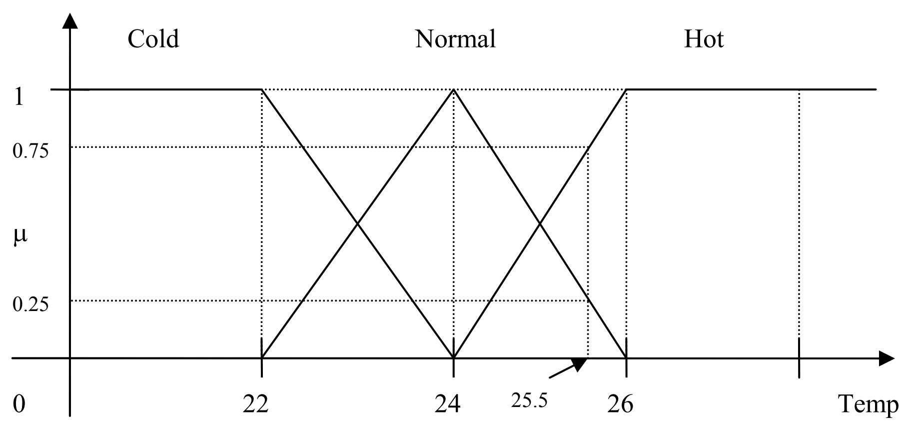

Figure 5.1 Membership Function of Fuzzy Sets 'Cold', 'Normal', and 'Hot'

Three fuzzy sets labeled as cold, normal and hot can be defined using three membership functions. The membership functions for the three fuzzy sets can be seen in Figure 5.1.

If the temperature (x) is 24 C, it definitely belongs to the normal fuzzy set ($\mu_{\text{normal}}(x) = 1$) but if the temperature is 25.5 C, it belongs to the normal fuzzy set with the degree of membership of 0.25 ($\mu_{\text{normal}}(x) = 0.25$) and it also belongs to the hot fuzzy set with the degree of membership of 0.75 ($\mu_{\text{hot}}(x) = 0.75$). The membership function can be triangular like the example above, trapezoidal, gaussian, bell, sigmoid etc. The correct form of a membership function will give the most efficient approximation of the specified system.

The knowledge base of fuzzy systems is in the form of IF THEN rules. If fuzzy logic is used for greenhouse temperature control, the rule is in the form of:

IF *the temperature is cold* **AND** *the heating control signal is small* **THEN** *heating control signal change is small positive*.

The first part of the rule "*the temperature is cold* **AND** *the heating control signal is small*" is called the antecedent or the premise while the last part of the rule "*heating control signal change is small positive*" is called the consequent or the conclusion. The rule above is activated if the inputs (temperature and heating control signal) belong to the fuzzy sets used in the rule. If the temperature belongs to the "cold" fuzzy set with the degree of membership function larger than zero and the heating control signal belongs to "small" fuzzy set to a degree larger than zero then the rule above is activated.

The word 'AND' in the rule represents the general classes of interception operators called triangular norm (t-norm). The most obvious member of the t-norm is the minimum operator. There are many other t-norm operators that can be used in place of the minimum operator such as algebraic product, bonded product, Dombi, Lukasiewicz etc. A complementary general union operator is called t-conorm and represented by the word 'OR'. The examples of t-conorm are maximum, algebraic sum, bounded sum and many others. A comprehensive discussion about the fuzzy set connective OR operator can be found in Pedrycz (Pedrycz and Gominde, 1998) and Nguyen (Nguyen and Walker, 2000). The AND operator combines the degree of membership of each fuzzy set in the rule to determine the firing strength.

The output of a fuzzy system (formally named fuzzy inference system) changes smoothly from one dominant rule to the other depending on the inputs combination. At any one time, one or two or three rules are activated at the same time with different firing strengths. The fuzzy inference system is a combination of local nonlinear functions that gives a combined output that changes smoothly (Jang, 1997). The OR operator combines the activated rules with different firing strengths to form an output. The output is still in a fuzzy form and can be defuzzified to get a crisp value. The process of determining the firing strength and then combining the output of activated rules are called fuzzy inference.

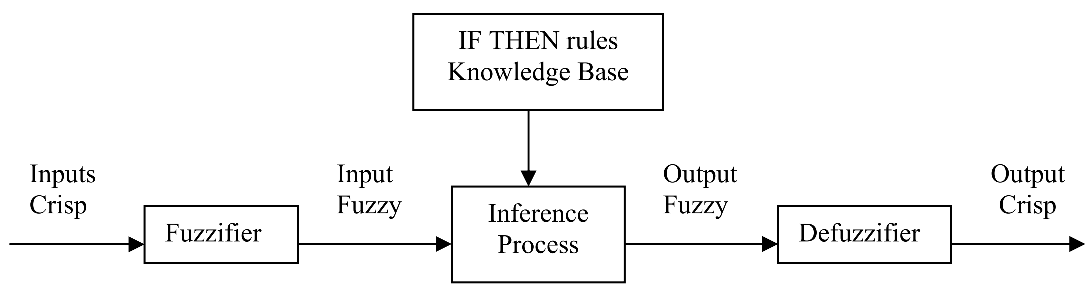

Figure 5.2 Block Diagram for a Fuzzy Inference System

The complete fuzzy inference system is as shown in Figure 5.2. The process of developing a fuzzy inference system involves: 1) Inputs selection, 2) Determining the shape of input membership functions, 3) Determining the number of fuzzy sets per input, 4) Defining the initial parameter value of membership functions, 5) Selecting suitable t-norm and t-conorm, 5) Building a IF THEN rules knowledge base and 6) Selecting the form of the output and deciding whether defuzzification is needed.

Tuning is needed to train the system for its desired purpose. It can be tuned by trial and error, by expert's input or by a learning algorithm. In a hydroponic plant production system, the exact knowledge of fault processes are not exactly known so both structure and tuning must be learned from experimental data. The Neuro-fuzzy model is the suitable choice to solve this problem.

5.2 Neuro-Fuzzy Model

Fuzzy systems and neural networks are complementary to each other. A fuzzy system is easy to comprehend because it uses linguistic terms and structure of IF THEN rules but it does not have a learning algorithm. Trial and error or expert knowledge is used in tuning the fuzzy system parameters and it can take a long time to finally find an acceptable system. Neural networks have many learning algorithms but it is extremely difficult to use a priori knowledge about the system. It is also almost impossible to explain the behavior of the neural system in a particular situation. A hybrid system with the best characteristics from both methods was developed and called a neuro-fuzzy system. A particular neuro-fuzzy system named ANFIS (Adaptive Network-based Fuzzy Inference System or more popularly named as Adaptive Neuro-Fuzzy Inference System) was proposed by Jang (1993).

5.2.1 The ANFIS Architecture

The ANFIS architecture is presented for a system with two inputs and a single output to better understand the performance of the structure. Consider a fuzzy inference system that has two inputs x and y and a singleton z as its output. For a first-order Sugeno model (Sugeno,1985), a common rule set with two fuzzy IF–THEN rules is as follows:

Rule 1 : IF x is A_1 and y is B_1 THEN $z = f_1 = p_0^1 + p_1^1 x + p_2^1 y$

Rule 2 : IF x is A_2 and y is B_2 THEN $z = f_2 = p_0^2 + p_1^2 x + p_2^2 y$

The reasoning mechanism for this Takagi-Sugeno model (Sugeno, 1985) is shown in Figure 5.3(a); the corresponding equivalent ANFIS architecture is shown in Figure 5.3(b). In the discussion below, the term $O_{j,i}$ represents the output of the i^{th} node in layer j, where nodes of the same layer have similar functions.

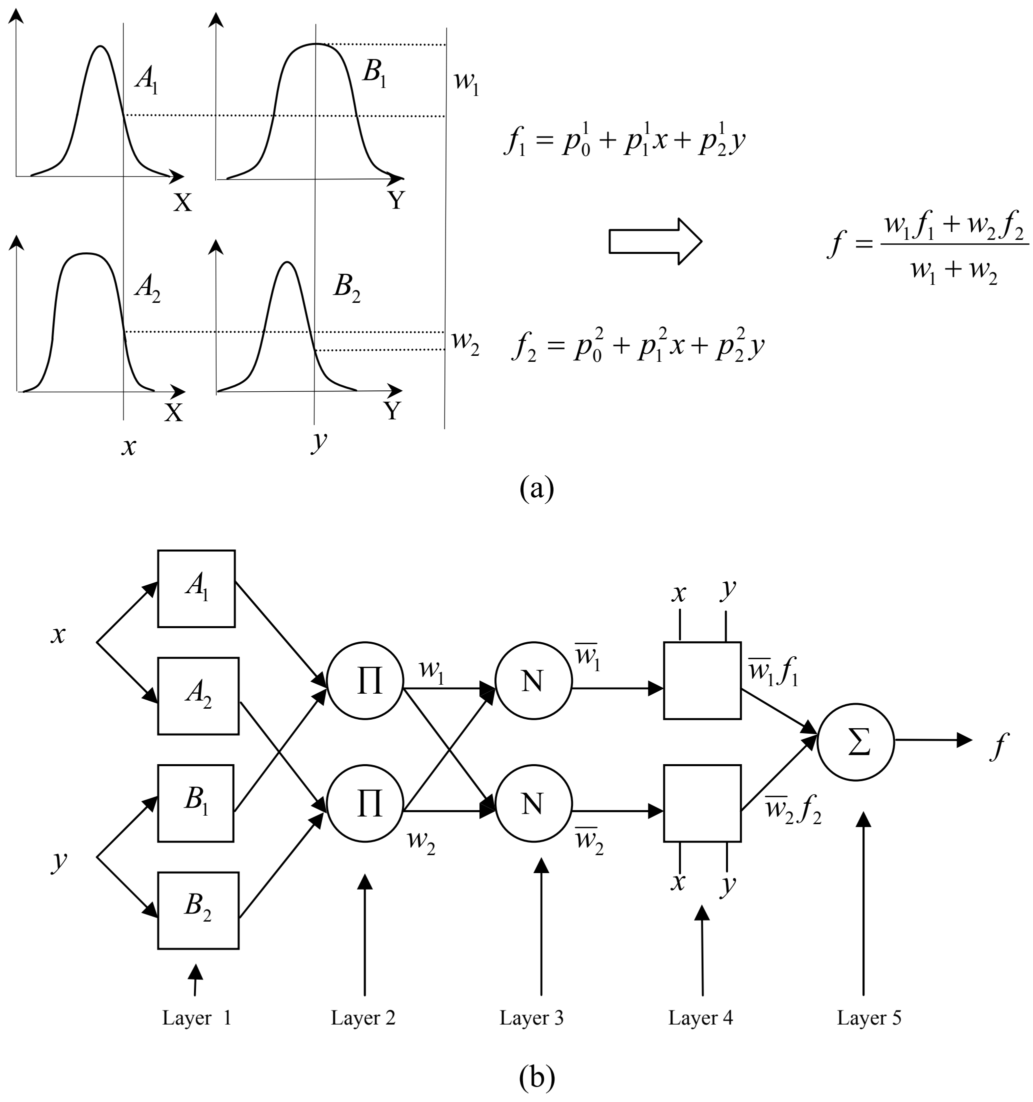

Figure 5.3 (a) Two rule two membership functionTS fuzzy model

(b) ANFIS equilavent of the TS model

Layer 1: Every node i in this layer is an adaptive node with a node function,

$$O_{j,i} = \mu_{A_i}(x), \qquad \text{for } i = 1,2 \quad \text{or}$$

$$O_{j,i} = \mu_{B_{i-2}}(y), \qquad \text{for } i = 3,4$$

where $\boldsymbol{x}$ (or $\boldsymbol{y}$) is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as "small" or "large") associated with the node. The membership function for A and B can be appropriate parameterized membership function such as the gaussian function :

$$\mu_A(x) = e^{-\frac{(x-m_i)^2}{2\sigma_i^2}}$$

where $\{m_i, \sigma_i\}$ is the parameter set

As the values of these parameters change, the function shape varies accordingly, thus exhibiting various forms of membership functions for fuzzy set A and B. Parameters in this layer are generally referred to as premise parameters.

Layer 2: Every node in this layer is a fixed node labeled Π whose output is the product of all the incoming signals

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \qquad \text{for } i = 1,2$$

Each node output represents the firing strength of a rule. In general, any other t-norm operators, which perform fuzzy AND can be used as the node function in this layer.

Layer 3: Every node in this layer is a fixed node labeled N. The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all the rules' firing strengths:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \qquad \text{for } i = 1,2$$

The outputs of this layer are usually referred as normalized firing strengths.

Layer 4: Every node i in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i\,(p_0^i + p_1^i x + p_2^i y), \qquad \text{for } i = 1,2$$

where $\overline{w}_i$ is the normalized firing strength from layer 3 and $\{ p_0^i,\ p_1^i, p_2^i \}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: This fixed layer, labeled Σ, gives the overall output as the summation of all incoming signals as follows:

$$\text{Overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

5.2.2 Hybrid-Learning Algorithm

It is shown from Figure 5.3 (a) that, when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. The output f in Figure 5.3 (b) can be written as

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \\ &= \overline{w}_1 (p_0^1 + p_1^1 x + p_2^1 y) + \overline{w}_2 (p_0^2 + p_1^2 x + p_2^2 y) \\ &= (\overline{w}_1) p_0^1 + (\overline{w}_1 x) p_1^1 + (\overline{w}_1 y) p_2^1 + (\overline{w}_2) p_0^2 + (\overline{w}_2 x) p_1^2 + (\overline{w}_2 y) p_2^2 \end{aligned}$$

where $p_0^1, p_1^1, p_2^1, p_0^2, p_1^2, p_2^2$ is linear in the consequent parameters. The consequent parameters can be obtained using this equation:

$$\begin{bmatrix} \overline{w}_1^{(1)} & \overline{w}_1^{(1)} x^{(1)} & \overline{w}_1^{(1)} y^{(1)} & \overline{w}_2^{(1)} & \overline{w}_2^{(1)} x^{(1)} & \overline{w}_2^{(1)} y^{(1)} \\ \overline{w}_1^{(2)} & \overline{w}_1^{(2)} x^{(2)} & \overline{w}_1^{(2)} y^{(2)} & \overline{w}_2^{(2)} & \overline{w}_2^{(2)} x^{(2)} & \overline{w}_2^{(2)} y^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{w}_1^{(n)} & \overline{w}_1^{(n)} x^{(n)} & \overline{w}_1^{(n)} y^{(n)} & \overline{w}_2^{(n)} & \overline{w}_2^{(n)} x^{(n)} & \overline{w}_2^{(n)} y^{(n)} \end{bmatrix} \begin{bmatrix} p_0^1 \\ p_1^1 \\ p_2^1 \\ p_0^2 \\ p_1^2 \\ p_2^2 \end{bmatrix} = \begin{bmatrix} r^{(1)} \\ r^{(2)} \\ \vdots \\ \vdots \\ \vdots \\ r^{(n)} \end{bmatrix}$$

where $[(x^{(k)}, y^{(k)}), d^{(k)}]$ are the k^{th} training data pair k = 1, 2,..,n and $\overline{w}_1^{(k)}$ and $\overline{w}_2^{(k)}$ are the outputs of layer 3 associated with the inputs ($x^{(k)}, y^{(k)}$).

Equation above can be expressed in matrix-vector form as:

$$\mathbf{Ax} = \mathbf{r}$$

Where $\mathbf{x} = \left[p_0^1, p_1^1, p_2^1, p_0^2, p_1^2, p_2^2\right]^T$, $\mathbf{r} = \left[r^1, r^2, ..., r^n\right]^T$ and **A** is a matrix formed by the elements $\overline{w}_1^{(k)}, \overline{w}_2^{(k)}, x^{(k)}, y^{(k)}$.

The above equation can be solved as

$$\mathbf{x}^* = (\mathbf{A}^\mathbf{T}\,\mathbf{A})^{-1}\,\mathbf{A}^\mathbf{T}\,\mathbf{r}$$

where $(\mathbf{A}^\mathbf{T}\,\mathbf{A})^{-1}\,\mathbf{A}^\mathbf{T}$ is the pseudoinverse of **A** if $(\mathbf{A}^\mathbf{T}\,\mathbf{A})^{-1}$ is non singular.

For a large size of training data set, an iterative method is preferable. $\mathbf{x}^*$ can be calculated recursively using the formula:

$$x_{(i+1)} = x_i + Q_{i+1}\, p_{i+1}^T\,(r^{(i+1)} - p_{i+1}\, x_i)$$

$$Q_{i+1} = Q_i - \frac{Q_i\, p_{i+1}^T\, p_{i+1} Q_i}{1 + p_{i+1} Q_i\, p_{i+1}^T} \qquad i = 0,1,2,\ldots\ldots n-1$$

$$\mathbf{x}^* = \mathbf{x}_n$$

with the initial conditions of

$$\mathbf{x}_o = 0 \text{ and } \mathrm{Q}_o = \gamma\,\mathrm{I}$$

where

γ is a positive large number and **I** is the identity matrix.

p_i is the i^{th} row vector of matrix **A**

$r^{(i)}$ is the i^{th} element of **r**

In the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameter are identified by the least squares method outlined above. In the backward pass, the signals that propagate backwards are the error signals and the premise parameters are updated by the gradient descent method. Table 5.1 summarizes the various activities during each pass (Jang, 1993).

Table 5.1 Parameter Update During The Forward and Backward Passes in Hybrid-Learning Procedure for ANFIS.

Signal flow direction	Forward Pass	Backward Pass
Consequent parameters	Least-squares estimator	Fixed
Premise parameters	Fixed	Gradient descent method
Signals	Node outputs	Error Signals

5.3 Input Selection

Suitable inputs must be chosen to develop the neuro-fuzzy fault detection system. Using as many related inputs as possible is desired in order to capture every possible symptom of the faults. A high number of symptoms makes the fault detection scheme more robust. On the other hand, a high number of inputs gives a complex FDI system, which needs a larger training data set and more training time and computational power. These requirements grow exponentially with every additional input variable. A balance is needed to optimize the system based on these two opposing requirements.

The neuro-fuzzy fault detection system is designed to detect and identify several faults whose symptoms are shown by different inputs so the selected inputs should be able to represent each fault sufficiently. Different faults have different time delays and time constants. These differences affect how many steps of previous sampling instants are needed for each fault. For example, with a five-minute sampling period, previous 5-minute, 10-minute and 15-minute sampled outputs are also needed in additional to current pH sensor output to detect pH control pump fault. Additional previous 20 and 25-minutes sampled outputs might be needed for pH sensor fault detection. An abrupt fault such as pH control pump fault can be detected faster than an incipient fault such as pH sensor fault. Abrupt fault also needs less previous sampling data.

These factors limit the number of faults that can be detected in one fault detection system with the finite amount of experiment data. Separate fault detection systems were developed for the biological faults and the actuator/sensor faults since both the incipient sensor faults and the biological fault need a high number of previous sampling data from different inputs.

With knowledge of the system dynamics, the variables involved in the faults can be found. When the circulation pump stopped working, the nutrient solution pH went up and the DO went down. The DO and pH controller tried to regulate the pH and the DO values according to the set points by adding increasing amounts of oxygen and concentrated acid with no result. Without the circulation, these additions had to rely on a slow diffusion process to reach the sensors. The controllers increased the control signal to maximum without any effect at the sensors for a long period. DO and DO control signals, and pH and pH control signal values were needed to detect the error. Their values from previous sampling steps were also needed. The number of previous sampling data needed was not known so several systems with different numbers of pH and DO previous sampling inputs were explored.

The pH control pump abrupt fault caused the pH value to increase despite the increasing pH control signal to keep the pH at the set point. The control signal eventually reached the maximum without any effect at the pH. The pH and pH control signal were needed as inputs for this fault detection.

The simulated EC sensor fault caused the EC value to drift slowly, first up and then down in a sinusoidal fashion. The fault definitely needed the EC measurement but other needed variables were unknown. In this experiment, a control signal was not available since EC was adjusted manually. It is assumed that the combination of input requirements from other faults was enough to develop symptoms for this fault.

The simulated pH sensor fault caused the pH value to drift down from the set point of 5.8 to 3.8 and then back up. Since the value of the pH never exceeded the set point in this experiment, the pH control signal was not directly affected. This kind of drifting was chosen since drifting upward caused the pH control to compensate by injecting some acid and by the time the experiment was over, the pH of the solution would be low enough to kill the roots. The pH value plus unknown interactions of inputs from other faults were assumed to give a specific pattern for this fault.

Inputs needed for the actuator/sensor fault detection systems were:

1. pH and its history data
2. pH control signal and its history data
3. DO and its history data
4. DO control signal and its history data
5. EC and its history data

The exact number of previous sampling steps that were needed for these five variables was not known so fault detection systems with the previous 2, 3, 4, 5, 6 and 7 sampling steps were developed and compared.

The solution temperature affected the metabolism of the plants and microorganisms in the solution which in turn determined the transpiration and nutrient absorption. Assuming the value of the temperature was changing slowly, no historical data was needed for this variable. The air temperature, light intensity and RH affected the plants and they were also included as input. Again their values were assumed to change slowly so no history data was included as input.

5.4 Membership Function

Once the inputs for the fault detection are selected, input membership functions must be determined. The gaussian membership function was selected for the neuro-fuzzy system since it has continuous derivability. This characteristic simplifies the learning process of the neuro-fuzzy system. The function is given by $\mu(x) = e^{-\frac{(x-m)^2}{2\sigma^2}}$. The Gaussian membership function is characterized by two parameters, namely m and σ. The desired Gaussian function can be obtained with the proper selection of the parameters m and σ. The parameter m represents the center of the Gaussian function and σ represents the width of the function.

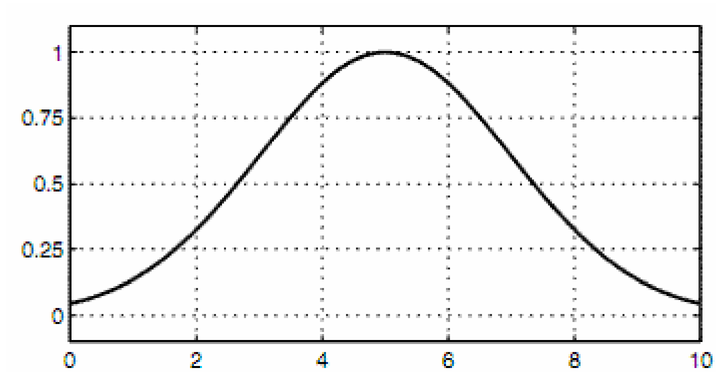

Figure 5.4 Gaussian membership function with $m = 5$ and $\sigma = 2$

5.5 Input Space Partitioning

The input space can be partitioned into grids by specifying the number of membership functions per input (figure 5.5 a and b). For example, a system with 11 inputs and 2 membership functions for each input will generate 2^{11} or 2048 grids where each grid represents one rule. If each membership function has 2 parameters in a Gaussian membership function, there are 2^{12} or 4096 parameters to be adjusted. This is the simplest way to build a fuzzy system and the most popular. The weakness of this approach is the large number of parameters that need to be optimized. Additional input increases the number of parameters exponentially. This problem is usually referred to as *the curse of dimensionality*.

A neuro-fuzzy system with the above configuration was developed for FDI at first for this dissertation. The number of parameters (4096) represented the maximum acceptable limit based on the amount of data available from experiments. The minimum amount of training data should be five times the number of parameters (Jang, 1997). The output from this neuro-fuzzy FDI system could not detect the desired faults very well. An insufficient number of inputs and membership functions caused bad performance of the system. Thus the grid partition method is suitable for fuzzy models with few input variables, which is not the case with FDI for hydroponic plant production system.

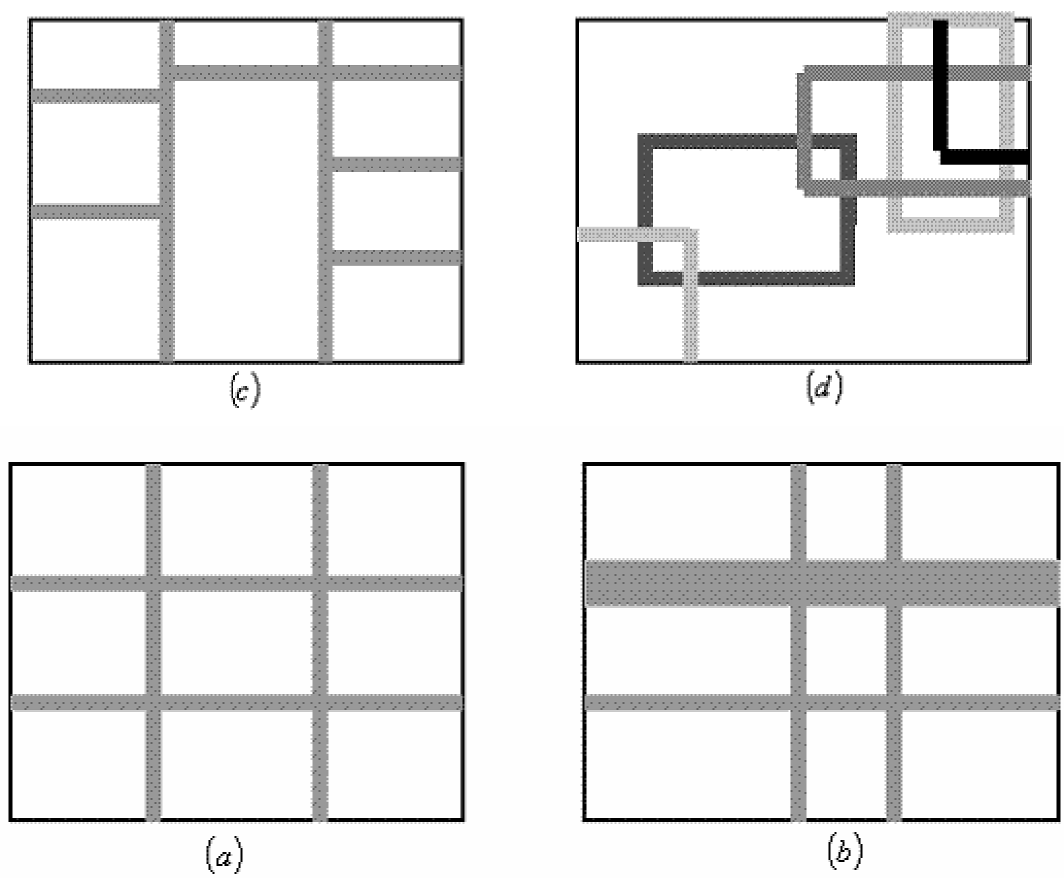

Figure 5.5 Space Partition
a) Uniform b) non-uniform grid partition c) tree partition d) scatter partition

Grid partitioning uniformly covers the whole input space. The monitored system usually does not have a uniform distribution of the input space and uses some subspace more often than others. A more efficient partitioning can be formed using this characteristic. Ignoring unused grids or lumping the seldom-used grids together into one reduces the number of grids and corresponding parameters in the neuro-fuzzy system.

A tree partition (figure 5.5 c) divides the input space into grids with different sizes by cutting the input space into different sized fuzzy regions. Frequently used subspaces are cut into small grids while rarely and unused subspaces are formed into a large grid. The tree partition solves the exponential increase in the number of parameters. The setback to this method is difficulty in determining the correct cut. More membership functions are needed to accommodate different sizes of subspaces.

Scatter partition/clustering is the most attractive choice (Figure 5.5 d). Instead of covering the whole space,, scatter partition tries to find subspaces that characterize the fuzzy region of the input space. It tries to cover the whole region of possible input vector occurrences. Scatter partition gives the most efficient partition with a smaller amount of computing time compared with other methods. The drawback of scatter partitioning is how the quality of the fuzzy system depends on the completeness of the data set in representing the whole operation region of the system. The scatter partition/clustering groups the input-output pairs into clusters and one fuzzy rule represents one cluster. The number of rules in the neuro-fuzzy system is equal to the number of clusters. Systems with different composition and number of clusters can be formed by varying parameters in the clustering algorithm.

5.6 Data Clustering

Data clustering algorithms are used to categorize and organize data. Then, these categorized data can be used for applications such as data compression, model building, etc. The clustering in the fuzzy system is useful for reducing the dimension of fuzzy system rules while still representing the overall system. Clustering partitions a data set into several clusters where each data points in a cluster has more similarity than the one among the clusters. In neuro-fuzzy systems, clustering is used to determine the initial locations and the number of IF-THEN rules. There are several clustering techniques that are used for this purpose and the most common ones are: K-means, fuzzy C-means, mountain clustering method and subtractive clustering.

5.6.1 K-Means Clustering

The *k-means* clustering is also known as the *hard c-means* clustering since a point belongs to only a particular cluster and not others. The opposite of this method is the fuzzy clustering which the data point can belong to several different clusters with different degree of memberships.

This clustering algorithm partitions a collection of n datapoints x_1, x_2, …, x_n, into c cluster. The cost function that minimizes the distance between the datapoints that belong in a cluster with cluster center v_i can be defined by

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} \left\| x_j - v_i \right\|^2$$

The clusters are defined by its cluster center and a $c \times n$ binary membership matrix U, where the element u_{ij} is 1 if the jth data point x_j belongs to the i^{th} cluster, and 0 otherwise. The process of determining the cluster center c_i, and the membership matrix U is iterative. The cluster centers are initialized randomly. The membership matrix U is then calculated as follows:

$$u_{ij} = \begin{cases} 1 \; if \; \|x_j - c_i\|^2 \leq \|x_j - c_k\|^2, \; for \; each \; k \neq i \\ 0 \; otherwise \end{cases}$$

The cost function is computed and the iteration can be stop if the value is below some tolerance or if the improvement over previous value is below some threshold. The cluster centers are updated using the new membership marix U as:

$$c_i = \frac{\sum_{j=1}^{n} u_{ij} x_k}{\sum_{j=1}^{n} u_{ij}}$$

The process is repeated again until satisfactory result is found or the number of iterations has been reached.

5.6.2 Fuzzy C-Means Algorithm (FCM) Approach

Conventional clustering algorithms locate a hard partition of a given data set where each entry of the data belongs to one partition or the other. On the other hand, the fuzzy clustering finds a soft partition of a given data set. Each entry of data can belong to a multiple of clusters. The degree of an entry in data to a cluster is given by a degree of membership. A widely used type of the fuzzy clustering algorithm is the fuzzy c-means or ISODATA (Dunn, 1973). James Bezdek has worked with the fuzzy pattern classification since his graduate years at Cornell University. He has developed it into one of the most popular clustering algorithm (Bezdek, 1973).

Dataset X with n data points: x_1, x_2, …, x_n, can be clustered into c fuzzy sets using the fuzzy c-mean clustering method. The criterion in most instances is to optimize an objective function that acts as a performance index of clustering. The end result of the fuzzy clustering can be expressed as a partition matrix U :

$U = u_{ij}$ with i = 1,…, c and j = 1,…,n

where u_{ij} is a numerical value between 0 and 1 and expresses the degree to which the datapoint $\boldsymbol{x}_j$ belongs to the i^{th} cluster. The objective function of the FCM algorithm takes the form of

$$J(u_{ij}, v_k) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \left\| x_j - v_i \right\|^2, \ m > 1$$

where m is the fuzziness factor, which influences the degree of fuzziness of the cluster partition. If m is a large number, a point with less membership in the cluster will have less influence on the calculation of the new cluster center. v_i is a cluster center the ith cluster $\{v_1, \ldots v_c\}$. To solve this minimization problem, the objective function is differentiated with respect to v_i (for fixed u_{ij}, $i = 1, 2, \ldots, c$; $j = 1, 2, \ldots, n$) and with respect to u_{ij}, (for fixed v_i, $i = 1, 2, \ldots, c$).

$$v_i = \frac{\sum_{j=1}^{n} (u_{ij})^m x_j}{\sum_{j=1}^{n} (u_{ij})^m}, \ i = 1,2, \ldots, c$$

$$u_{ij} = \sum_{k=1}^{c} \left[\left(\frac{\left\| x_j - v_i \right\|^2}{\left\| x_j - v_k \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}, \ i = 1, 2, \ldots, c; j = 1, 2, \ldots, n.$$

After the number of clusters c ($2 \le c \le n$) and fuzziness factor have been determined, the initial partition matrix U is chosen randomly. Cluster centers and the partition matrix can be calculated iteratively from the above equations. If the difference of the previously calculated center and/or partition matrix and the current value is less the predetermined threshold, the process can be stopped.

5.6.3 The Mountain Clustering Algorithm

The mountain clustering method is a grid-based method for identifying the approximate locations of the cluster centers (Yager, 1994). Unlike fuzzy c-means, this method does not require a predetermination of the number of clusters. Grid points on the data space provide the potential cluster centers. A finer grid increases the number of potential cluster centers but it also increases the computation required. The grid is generally evenly spaced, but it is not required. Uneven spaced grids that reflect the prior knowledge of the data space can be formed.

Grid point selection for a cluster center is based on the mountain function. The height of the mountain function at a grid point g is equal to

$$M(g) = \sum_{i=1}^{n} \exp\left(-\frac{\|g - x_i\|}{2\sigma^2} \right)$$

where x_i is the i^{th} data point and σ is an application specific constant. The closer the data point x_i to the grid point, the more it contributes to the height of the mountain function. The value of the mountain function reflects the density of data points in the vicinity of each grid point. The higher the mountain function value at a grid point the larger it's potential for being a cluster center. The grid node with the highest score of the mountain function is selected and becomes the first cluster center v_1. The next cluster center could not be selected yet since the first cluster center is usually surrounded by a number of grid points which also have high density values. The effect of the first center must be eliminated by sequentially destructing the mountain function. In order to do so a revised mountain function is formed:

$$M_{new}(g) = M(g) - M(v_1)\exp\left(-\frac{\|g - v_1\|^2}{2\beta^2} \right)$$

After the subtraction, the new mountain function value at v_1 is zero and its effect on surrounding points is eliminated. The second cluster center then can be selected from the grid point with the highest value of the new mountain function. This process is repeated until the new mountain function value is less than a stopping constant.

5.6.4 Subtractive Clustering

The mountain clustering method is simple and very effective in finding cluster centers that can be the base of fuzzy system membership function. However, the number of calculations required grows exponentially with the dimensions of the data set. For data set of 3 variables and 10 grid points for each variable, 1,000 points must be evaluated. Adding another variable to the data set multiplies the grid points by 10 or 10,000 grids.

A variation of the mountain method called subtractive clustering solved this problem (Chiu, 1994). Instead of using grid points, data points are used as candidates of the cluster centers. By doing this, the computation needed for calculation is proportional to the number of data points and independent of the dimension of the problem (the variables). This rough calculation of the cluster centers is particularly suitable if the clustering method is used to find the initial structure of a fuzzy system that will be optimized later by the neural network learning algorithm.

For a data set of n data points, a density measure at data point x_i is defined as

$$D_i = \sum_{j=1}^{n} \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{\left(r_a / 2 \right)^2} \right)$$

where r_a is a positive constant. A data point will have a high density value if it has many neighboring data points.

As in the mountain method, the data point with the highest density measure is selected as the first cluster center v_1. The next step is to eliminate the influence of the first cluster center to the surrounding data points which also have high density values. The density measure of each data point is revised as

$$D_{i(new)} = D_i - D_{v_1} \exp\left(-\frac{\left\| x_i - x_{v_1} \right\|^2}{\left(r_b / 2 \right)^2} \right)$$

where r_b is a positive constant. The density measure of data points in the neighborhood of the cluster center v_1 is reduced and the one at the first cluster center is zero. The effect of the first cluster center on surrounding points is eliminated. The constant r_b defines a neighborhood that has significant reduction in density measures after the revision. The constant r_b is usually larger than r_a to prevent closely spaced cluster centers. Generally r_b is chosen to be equal to 1.5 r_a.

The point with the highest density measure is selected again as the next cluster center. This process is iterated until the highest density measure is lower than a predetermined stopping constant or sufficient number of cluster centers has been determined. The result can be used for developing the Takagi-Sugeno fuzzy model. Cluster centers v_i are the fuzzy system rules. The degree of fulfillment of the fuzzy rule i is defined by

$$\mu_i = \exp\left(-\frac{\|x - v_i\|^2}{(r_a / 2)^2} \right)$$

After completed these procedure, a more accurate system can be constructed using optimization scheme like the gradient descent algorithm.

5.7 Multi Level Value Neuro-fuzzy Fault Detection System

The neuro-fuzzy fault detection and identification system developed in this research tried to find the direct connection between the combination of input variables and the faults themselves. The neuro-fuzzy fault detection system in this research utilized one output to detect and identify multiple faults. Different faults are represented by different output values. The output value of 1 is reserved for a normal condition, the value of 2 for pH control pump fault, the value of 3 for circulation pump fault, the value of 4 for pH sensor fault and the value of 5 for EC sensor fault. Using a single multi level value output simplifies the model and reduces the computational time needed to optimize each model. The use of only one output to detect several faults with widely different dynamics is the ultimate test for a neuro-fuzzy system since it combines both the residual generation and the residual analysis stages into one.

A similar multi level value neuro-fuzzy FDI system was also planned for biological faults but the signals of many simulated biological faults symptoms were too small compared with the noise of the monitored system. The exception to this problem was a transpiration fault where leaves of each plant were covered by a plastic bag. This treatment simulated a biological fault that drastically affecting transpiration in the plants. The neuro-fuzzy biological FDI system has a dedicated output for the transpiration fault.

CHAPTER 6

NEURO-FUZZY BIOLOGICAL FAULT DETECTION AND IDENTIFICATION SYSTEM

6.1 Biological Faults in The Hydroponic System.

Biological faults in the hydroponic system can be categorized into shoot zone faults and root zone faults. Two different types of experiments from each category were performed from November 2000 to June 2001. Bruising and cutting the leaves of lettuce plants were performed to simulate shoot zone faults. Removing the plants from the water for 5 minutes and covering the whole leaves by a plastic bag simulated root zone faults.

There were no significant changes in DO, temp, pH, EC and weight changes for the shoot zone faults. Deviations in parameters caused by leaves bruising were too small compared with the noise in the system. Experiments with cutting leaves showed unexpected result since the evapotranspiration was not reduced at all. Water loss from the wound gave up water comparable to the normal plant transpiration.

The first root zone fault experiments also cannot be detected, signaling a much bigger disturbance must be ministered. Covering the whole leaves of the largest plants (ages of 23, 25, 27 days) with plastic bags showed a positive deviation in the transpiration rate. With this development, the biological multilevel value FDI system becomes the single value transpiration FDI system. The FDI system output was trained to have a value of 0 for normal and 1 for transpiration faults.

6.2 Neuro-Fuzzy FDI Specifications

Neuro-fuzzy biological FDI systems with 5 and 10 minutes interval data were developed. Systems with five-minute interval data were developed since the data can be used directly from the data file. Construction of systems with ten-minute interval data was intended to explore the noise reduction in the FDI system by data averaging.

The FDI systems process the current sample of air temperature, light intensity, relative humidity (RH), nutrient temperature, pH, DO, EC, pH control signal, DO control signal, weight rate and previous weight rate samples. The biological FDI with 24 inputs has previous 14 weight rate samples and the biological FDI with 29 inputs has the previous 19 weight rate samples.

The subtractive clustering was used to extract neuro-fuzzy fault rules from input-output data. Range of influence (*roi*) coefficients in the clustering method determine how many cluster centers formed. Values between 0.2 and 0.5 are recommended (Chu, 1994). Several *roi* values were used to form the neuro-fuzzy systems. A small *roi* means a short range of influence of the cluster center and a large number of cluster centers formed. The number of cluster center determines the number of fuzzy rules.

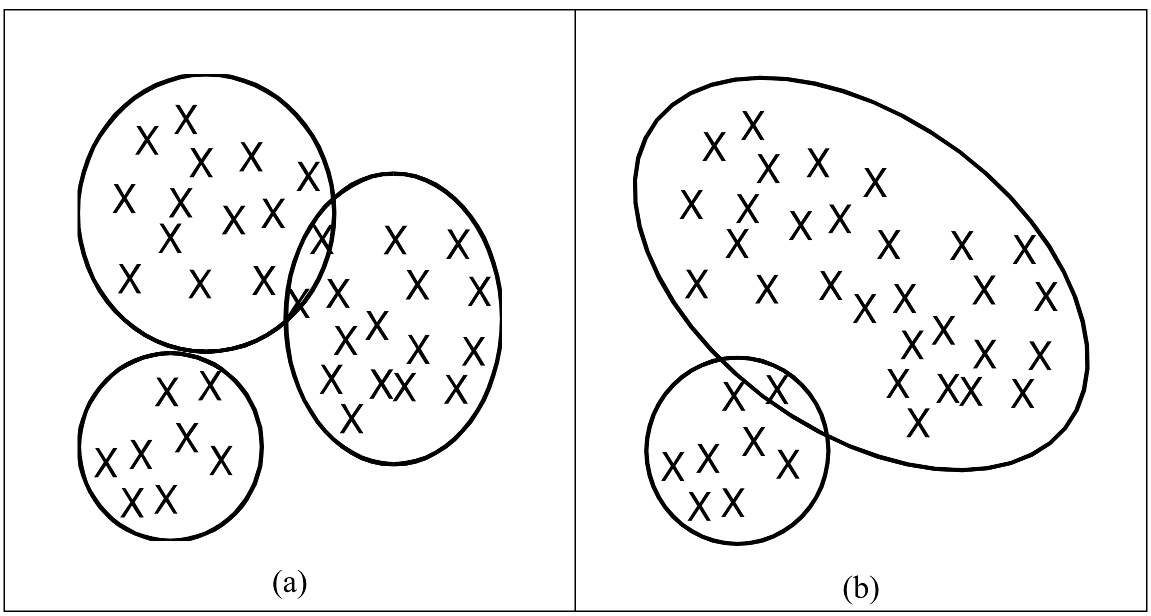

Figure 6.1 The Effect of *roi* (Range of Influence Constant) to The Number of Formed Clusters for a Simple Two Dimensional Dataset

(a) *roi* of 0.25 (b) *roi* of 0.45

For example, a subtractive clustering algorithm with a *roi* value of 0.25 forms 23 cluster centers (rules) for the FDI system with 29 inputs. Graphics in Figure 6.1 illustrate the effect of *roi* to the number of clusters for a simple two dimensional data set.

6.3 Data Sets

Data was divided into training and testing data sets. There was an effort to choose datasets that covered the whole experiment period. Training data sets for the neuro-fuzzy biological FDI systems are shown in Table 6.1.

Table 6.1 Biological Fault Training Data Sets

Data File Start	Data File End	Dataset Type
12/15/00 12:01 am	12/18/00 12:00 am	Normal Train
02/26/01 12:01 am	03/03/01 12:00 am	Normal Train
03/25/01 12:01 am	04/04/01 11:55 pm	Normal Train
04/11/01 12:02 am	04/12/01 11:58 pm	Normal Train
02/20/01 12:02 am	02/25/01 06:01 am	Transpiration Fault
04/05/01 12:00 am	04/10/01 05:59 am	Transpiration Fault
04/13/01 12:03 am	04/18/01 05:59 am	Transpiration Fault
04/25/01 12:02 am	04/30/01 06:02 am	Transpiration Fault

All trained systems were tested with 3 data sets as shown in Table 6.2. The first two data sets are the transpiration fault testing data and the last is for the normal condition.

Table 6.2 Biological Fault Testing Data Sets

Data File Start	Data File End	Test Dataset Type	Test #
03/06/01 12:04 am	03/11/01 11:59 pm	Transpiration Fault	Test 1
05/23/01 12:03 am	05/28/01 12:00 pm	Transpiration Fault	Test 2
04/19/01 12:03 am	04/22/01 12:37 pm	Normal test	Test 3

6.4 Training Results

FDI systems constructed by the subtractive clustering method were trained further using training data from fault experiments. Each system was trained for 5 epochs with each epoch consists of 500 iterations (a total of 2500 iterations). An epoch is a batch of training iterations. The 5-minute interval systems training results are shown in Table 6.3.

Several things can be seen directly from the table 6.3. Neuro-fuzzy FDI systems with smaller initial error generally continue to have smaller error at the end of the training. For example, the FDI system with 39 input and 25 rules had the least initial error and after 5 epochs of training it still had the least error compared with other systems.

Systems with more inputs usually have the least training error. For example, the 39-input systems have less error than the 34-input systems. Systems with more inputs have more degrees of freedom in modeling the monitored process, and less error.

Systems with more rules can capture the dynamics of faults better than those with fewer rules. Systems with 39 inputs and 25 rules had a training error of 0.4612 while the one with 7 rules had 0.6371 as the training error. Again, additional rules give more modeling freedom for the system.

Additional training reduces training error. The first epoch of training reduces the error the most while the last epoch reduces it the least. The number of training epochs was limited to 5, because further training did not give any significant improvement. All systems were trained to the same number of epochs so they have the same state of training for comparison. Figure 6.2 shows error trend for every epoch of training

Table 6.3 Training Results of Biological Fault Detection Systems with 5-Minute Interval

Roi	**Rules**	**The 1st Epoch**		**The 2nd Epoch**		**The 3rd Epoch**		**The 4th Epoch**		**The 5th Epoch**	
		Start Error	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**
						24 INPUTS					
0.25	24	0.6848	0.5365	0.5365	0.5285	0.5285	**0.5191**[d]	0.5191	**0.515**[d]	0.515	**0.5124**[d]
0.3	17	0.7098	0.5647	0.5647	0.5561	0.5561	0.5549	0.5549	0.5551	0.5549	0.5546
0.4	10	0.7882	0.6554	0.6554	0.6535	0.6535	0.6525	0.6525	0.6518	0.6518	0.6512
0.5	7	0.8008	0.6781	0.6781	0.6731	0.6731	0.666	0.666	0.6654	0.6654	0.6652
						29 INPUTS					
0.25	23	**0.6693**[d]	**0.5332**[d]	0.5332	**0.5276**[d]	0.5276	0.5239	0.5239	0.522	0.522	0.5213
0.3	18	0.6842	0.5476	0.5476	0.5421	0.5421	0.5412	0.5412	0.5417	0.5412	0.541
0.4	11	0.7553	0.6224	0.6224	0.6158	0.6158	0.6123	0.6123	0.6108	0.6108	0.6094
0.5	7	0.7854	0.6589	0.6589	0.652	0.652	0.6512	0.6512	0.6506	0.6506	0.6502
						34 INPUTS					
0.25	24	**0.649**[b]	**0.5096**[b]	0.5096	**0.4893**[b]	0.4893	**0.4818**[b]	0.4818	**0.4807**[b]	0.4807	**0.4801**[b]
0.3	18	0.6758	0.5426	0.5426	0.5347	0.5347	0.5248	0.5248	0.5243	0.5243	0.524
0.4	11	0.7416	0.6029	0.6029	0.5993	0.5993	0.5965	0.5965	0.5956	0.5956	0.5952
0.5	7	0.7755	0.6477	0.6477	0.6428	0.6428	0.642	0.642	0.6415	0.6415	0.6412
						39 INPUTS					
0.25	25	**0.6313**[a]	**0.4814**[a]	0.4814	**0.4701**[a]	0.4701	**0.465**[a]	0.465	**0.4628**[a]	0.4628	**0.4612**[a]
0.3	19	**0.6612**[c]	**0.5247**[c]	0.5247	**0.5161**[c]	0.5161	**0.5131**[c]	0.5131	**0.5095**[c]	0.5095	**0.5066**[c]
0.4	11	0.732	0.607	0.607	0.6011	0.6011	0.5914	0.5914	0.5883	0.5883	0.5877
0.5	7	0.768	0.6451	0.6451	0.6393	0.6393	0.6385	0.6385	0.6379	0.6379	0.6371

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

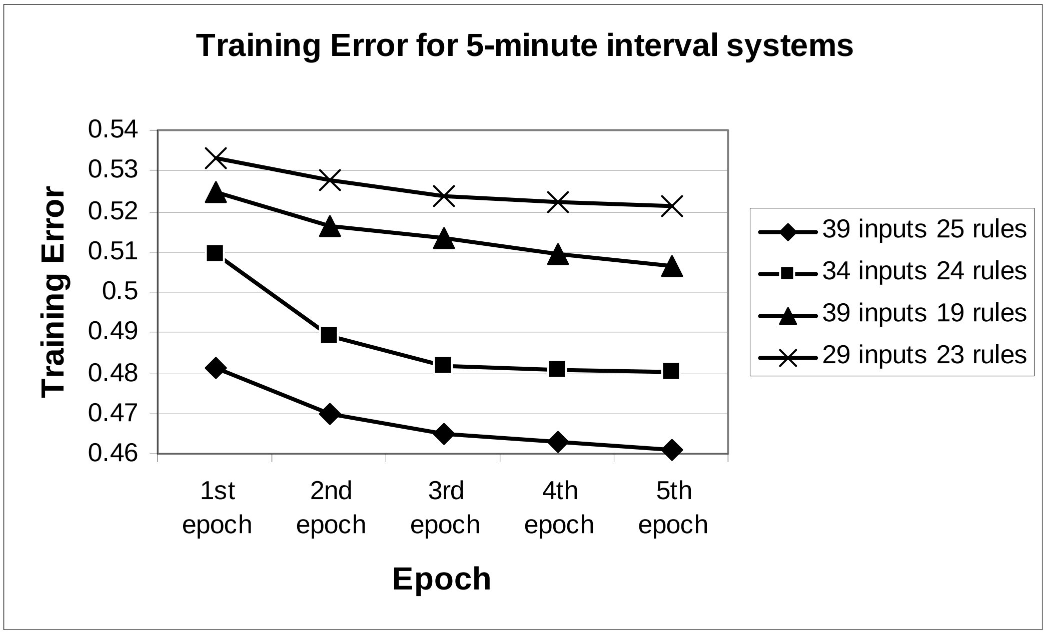

Figure 6.2 Training Error Trend for 5-Minute Interval Systems

Training results for 10-minute interval systems are shown in Table 6.4. Compared with 5-minute interval systems, 10-minute interval systems have more rules. This means that the subtractive clustering algorithm found more cluster centers for the 10-minute training data set. Averaging data points usually reduces the high frequency noise in the data and the clusters are more separated from each other. The resulting systems perform better than those with 5-minute intervals. The best 10-minute interval system has nearly half the amount of error compared to the best 5-minute interval system. The training error trends for 10-minute interval systems are shown in Figure 6.3

Table 6.4 Training Results of Biological Fault Detection Systems with 10-Minute Interval

		The 1st Epoch		The 2nd Epoch		The 3rd Epoch		The 4th Epoch		The 5th epoch	
Roi	**Rules**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**
					24 INPUTS						
0.32	35	0.6155	0.4098	0.4098	**0.3846** [d]	0.3846	**0.3716** [d]	0.3716	**0.3631** [d]	0.3631	**0.3562** [d]
0.35	26	0.6363	0.4451	0.4451	0.4252	0.4252	0.4173	0.4173	0.4122	0.4122	0.4083
0.4	16	0.6698	0.5328	0.5328	0.5158	0.5158	0.5082	0.5082	0.5033	0.5033	0.499
					29 INPUTS						
0.33	39	**0.5842** [b]	**0.3719** [b]	0.3719	**0.351** [b]	0.351	**0.3304** [b]	0.3304	**0.3202** [b]	0.3202	**0.3157** [b]
0.35	30	**0.613** [d]	**0.3979** [d]	0.3979	0.3849	0.3849	0.3778	0.3778	0.3725	0.3725	0.3682
0.4	22	0.6192	0.4645	0.4645	0.4463	0.4463	0.4358	0.4358	0.4293	0.4293	0.425
					34 INPUTS						
0.38	39	**0.5611** [a]	**0.322** [a]	0.322	**0.3046** [a]	0.3046	**0.2949** [a]	0.2949	**0.2879** [a]	0.2879	**0.2824** [a]
0.4	32	**0.5861** [c]	**0.3831** [c]	0.3831	**0.3598** [c]	0.3598	**0.3456** [c]	0.3456	**0.3391** [c]	0.3391	**0.3346** [c]
0.45	22	0.6133	0.4598	0.4598	0.4504	0.4504	0.4455	0.4455	0.4414	0.4414	0.4381

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

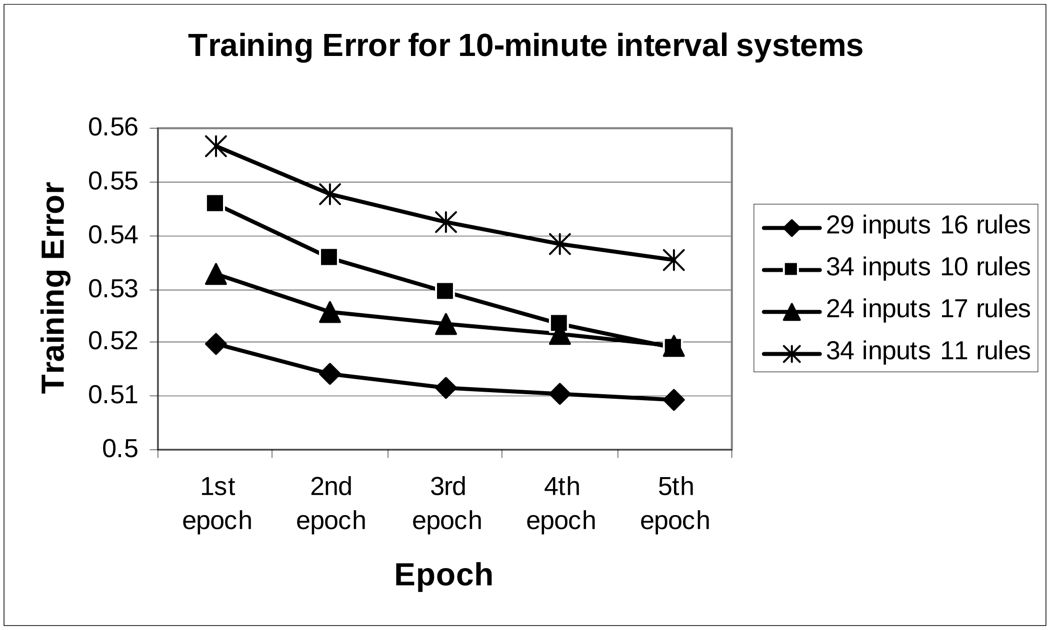

Figure 6.3 Training Error Trend for 10-Minute Interval Systems

6.5 FDI System Performance Definitions

FDI system performance can be evaluated by detection time and correct classification of faults. Detection time is the time needed by the FDI system to detect the occurrence of the fault in the monitored system. The FDI system makes a correct classification if the system output shows the correct level for the intended fault after the fault detection. Misclassification does not include discrepancies at the output during the detection time. Fault level categorization is shown in Figure 6.4.

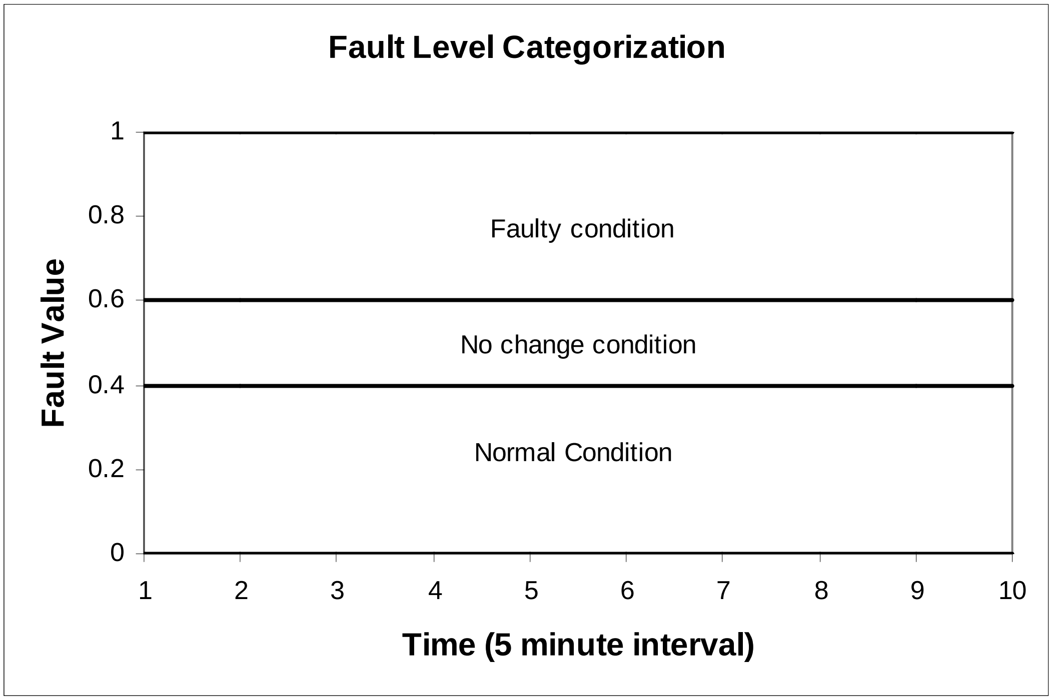

Figure 6.4 Fault Level Categorization

The output of the fault detection system gradually goes from a normal condition represented by a value of 0 to a faulty condition with a value of 1. Any response above 1 is assumed to be 1 and represents the faulty condition. Any response below 0 is assumed to be 0 and represents the normal condition.

The area between 0 and 1 can be divided into three parts as shown in Figure 6.4. The value between zero and 0.4 is defined as normal, between 0.4 and 0.6 is defined as no change from the previous condition. And between 0.6 and 1 is defined as a faulty condition. For example at time $t-1$ (one sampling step before the current sampling instant), the response y is 0.7, representing faulty condition. At time t, $y=0.5$, according to this definition, the condition at time t is still faulty. If, at $t+1$, $y= 0.35$ then the response has changed to a normal condition.

6.6. Testing Results

The training results confirm the ability of the neuro-fuzzy systems to categorize faults according to the training data. It is still possible for the neuro-fuzzy systems to be over trained. Overtrained is the condition where the neuro-fuzzy system can follow the training data very closely but respond very poorly to a new data set. Separate data sets are used to test whether the system is over trained. The over trained system will not process the testing data correctly

The testing results of the Biological Fault Detection systems with 5-minute interval data are shown in Table 6.5 and systems with 10-minute interval data are shown in Table 6.6. Most of the systems can detect the intended faults. Systems with least errors in training are also performed well in testing, proving they are not over trained. The test errors for the best four responses are all below 0.3 for each of the tests.

Systems with the combination of a high number of rules and inputs have the smallest errors. Three of the four systems with smallest error have the highest number of rules with 25, 24, and 23 rules. Systems with 39 inputs and 19 rules rank third while system with 29 inputs and 23 rules ranks fourth. It seems that the number of inputs is more important than the number of rules.

The errors in Table 6.5 and Table 6.6 show that the 10-minute interval is a better time step than the 5-minute interval. Testing errors of the best four 10-minute interval systems for the first test are between 0.189 and 0.237. These are significant improvements compared with systems with 5-minute interval where the best error is 0.273. The second test results are even better. The best error is 0.072 compared with 0.196 for the best error of the 5-minute interval systems. That means the error is less than half of the best 5-minute interval system. The third test result best error is 0.139, nearly half the best error for 5 minute interval systems of 0.269.

Table 6.5 Testing Results of Biological Fault Detection Systems with 5-Minute Interval

Roi	Rules	Training Error	Testing Error			
			Test 1	Test 2	Test 3	Error Sum
			24 INPUTS			
0.25	24	**0.5124** [d]	0.308460705	**0.20248051** [c]	0.278923034	0.789864249
0.3	17	0.5546	0.324716565	0.221644479	0.293387448	0.839748492
0.4	10	0.6512	0.415115559	0.238148404	0.344361686	0.997625649
0.5	7	0.6652	0.407668653	0.261664896	0.380455159	1.049788708
			29 INPUTS			
0.25	23	0.5213	**0.30491074** [d]	0.206859032	**0.274812928** [c]	**0.7865827** [d]
0.3	18	0.541	0.329627311	0.218513679	0.279370008	0.827510997
0.4	11	0.6094	0.356333331	0.245778224	0.332142552	0.934254107
0.5	7	0.6502	0.38912825	0.253844917	0.354930749	0.997903916
			34 INPUTS			
0.25	24	**0.4801** [b]	**0.281143788** [b]	**0.19651883** [b]	0.280049801	**0.757712418** [b]
0.3	18	0.524	0.326984888	0.211080144	**0.277461864** [d]	0.815526896
0.4	11	0.5952	0.340228987	0.242100911	0.338398463	0.920728362
0.5	7	0.6412	0.370739567	0.255570979	0.350072279	0.976382824
			39 INPUTS			
0.25	25	**0.4612** [a]	**0.272664835** [a]	**0.19639848** [a]	**0.269275274** [a]	**0.738338588** [a]
0.3	19	**0.5066** [c]	**0.290587246** [c]	**0.204278027** [d]	**0.272020062** [b]	**0.766885335** [c]
0.4	11	0.5877	0.334768592	0.23082643	0.323844354	0.889439377
0.5	7	0.6371	0.376257849	0.250696988	0.35803487	0.984989707

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

Table 6.6 Testing Results of The Biological Fault Detection Systems with 10-Minute Interval

Roi	Rules	Training Error	Testing Error: Test 1	Testing Error: Test 2	Testing Error: Test 3	Testing Error: Error Sum
24 INPUTS						
0.32	35	**0.3562** [d]	**0.234897259** [c]	0.135800372	**0.179293787** [c]	**0.549991418** [d]
0.35	26	0.4083	0.272624582	0.131572433	0.217903591	0.622100607
0.4	16	0.499	0.345997333	0.153457275	0.254178781	0.753633389
29 INPUTS						
0.33	39	**0.3157** [b]	**0.207358717** [b]	**0.102819571** [d]	**0.153521143** [b]	**0.463699431** [b]
0.35	30	0.3682	0.2544858	0.115566548	0.18593746	0.555989809
0.4	22	0.425	0.354050497	0.176326231	0.330779066	0.861155794
34 INPUTS						
0.38	39	**0.2824** [a]	**0.189306555** [a]	**0.087423926** [b]	**0.138518606** [a]	**0.415249087** [a]
0.4	32	**0.3346** [c]	**0.236912814** [d]	**0.071872805** [a]	**0.179983194** [d]	**0.488768813** [c]
0.45	22	0.4381	0.29590013	**0.093677391** [c]	0.232099377	0.621676898

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

Table 6.7 Testing Results of The Biological Fault Detection System with Various Stage of Training

roi	rules	Training Error	Epoch #	Testing Error: test 1	Testing Error: test 2	Testing Error: test 3	Testing Error: Error Sum
0.38	39	**0.2824**	1	0.218971171	0.096143163	0.169036908	0.484151241
			2	**0.206731425** [d]	**0.091506611** [d]	**0.155317747** [d]	**0.453555783** [d]
			3	**0.199110828** [c]	**0.089544775** [c]	**0.148376336** [c]	**0.437031939** [c]
			4	**0.193651243** [b]	**0.088378694** [b]	**0.14272211** [b]	**0.424752047** [b]
			5	**0.189306555** [a]	**0.087423926** [a]	**0.138518606** [a]	**0.415249087** [a]

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

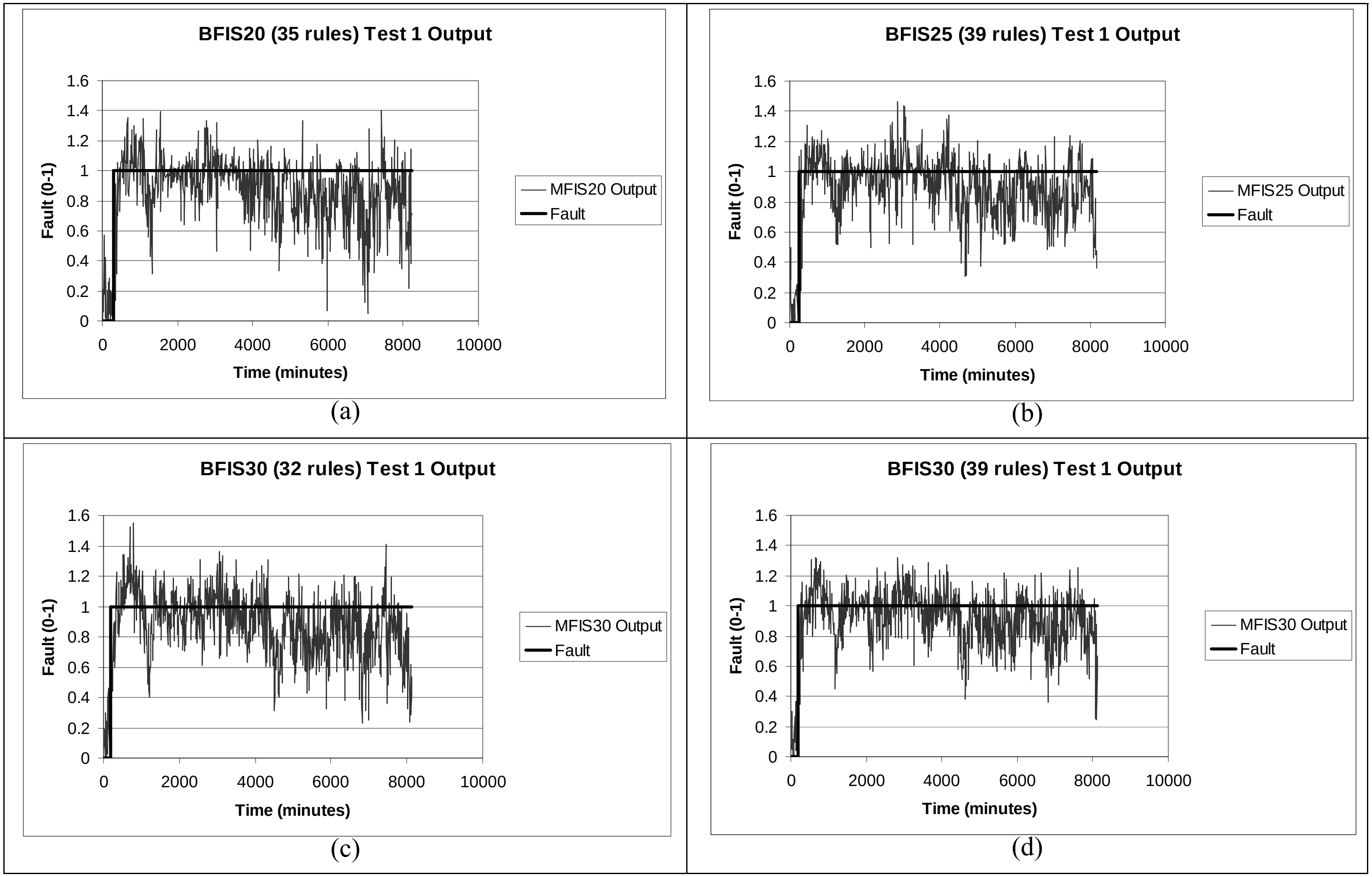

Figure 6.5 Test 1 (Transpiration Fault) Output for The Biological Fault Detection System with 10-Minute Interval

Table 6.7 shows how the fault detection system evolves with training. This particular example is the 10-minute interval system with 34 inputs and 39 rules. The test error is reduced for each test when additional training is applied to the system up to epoch number 5. This solidifies the conclusion that the fault detection system is not overtrained since both the training error and testing error are reduced with additional training.

Although testing errors in Table 6.5 and Table 6.6 are good indicators of the FDI systems ability in processing the data set, they do not show the dynamic of the response. For this purpose, the best four 10-minute interval systems response charts are shown for each test.

Figure 6.5 shows the results for transpiration fault test 1. All FDI systems have noisy responses. The noise caused delays in the transpiration fault detection. The detection time for all FDI system is about for 50 minute. The summary of FDI systems performance can be seen in Table 6.8. The misclassification decreases as the number of inputs increases for the FDI systems.

Table 6.8 FDI Systems Performances of Test 1 (Transpiration Fault)

FDI Systems	Detection Time	Misclassification	Correct Classification
24 inputs and 35 rules	50 minutes	1.80%	98.20%
29 inputs and 39 rules	50 minutes	1%	99%
34 inputs and 32 rules	50 minutes	2%	98%
34 inputs and 39 rules	50 minutes	0.60%	99.40%

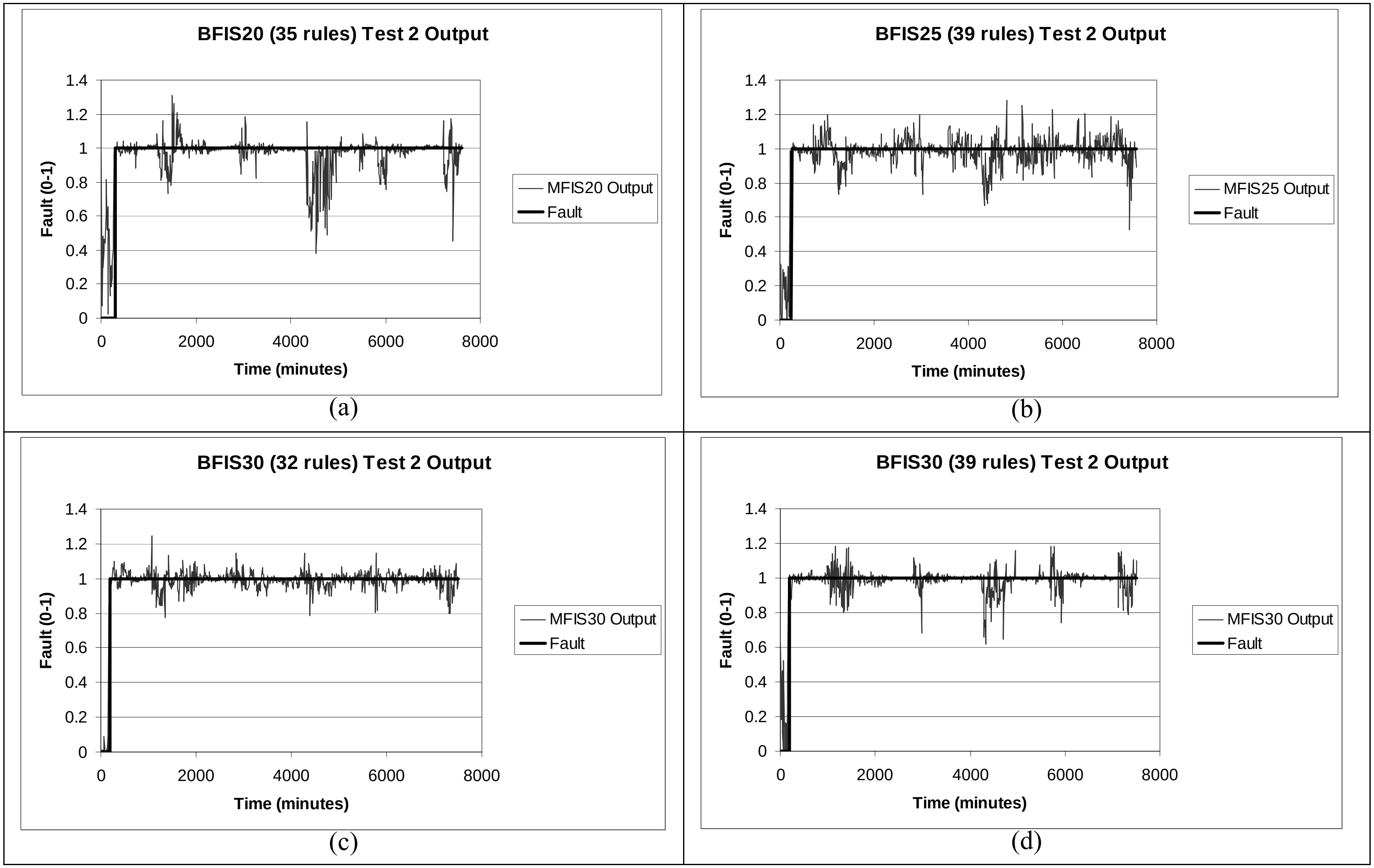

Figure 6.6 Test 2 (Transpiration Fault) Output for Biological Fault Detection Systems with 10-Minute Interval

The second test responses of the four best systems with 10-minute intervals are shown in Figure 6.6. The responses are very clean with very little noise compared with test 1. This means the faulty condition pattern was represented well in the training data sets. There is periodic noise in every response caused by maintenance disturbances about every two days (about 290 points) and harvesting for another overlapping 2 days period.

The summary of the FDI systems performance is shown in Table 6.9. All FDI systems recognized the fault almost immediately. The process of covering the leaves of the lettuce plants took about 30 minutes and the fault starting point was defined when the covering activity finished. The FDI started recognizing the fault when the covering process happened.

Table 6.9 FDI Systems Performances of Test 2 (Transpiration Fault)

FDI systems	Detection time	Misclassification	Correct classification
24 inputs and 35 rules	10 minutes	0.90%	99.10%
29 inputs and 39 rules	10 minutes	0%	100%
34 inputs and 32 rules	10 minutes	0%	100%
34 inputs and 39 rules	10 minutes	0%	100%

Misclassification is 0% for three out of the four FDI systems. This is an excellent result for a slowly happening fault in the monitored system. External disturbances caused by maintenance and harvesting can be overcome by the FDI systems.

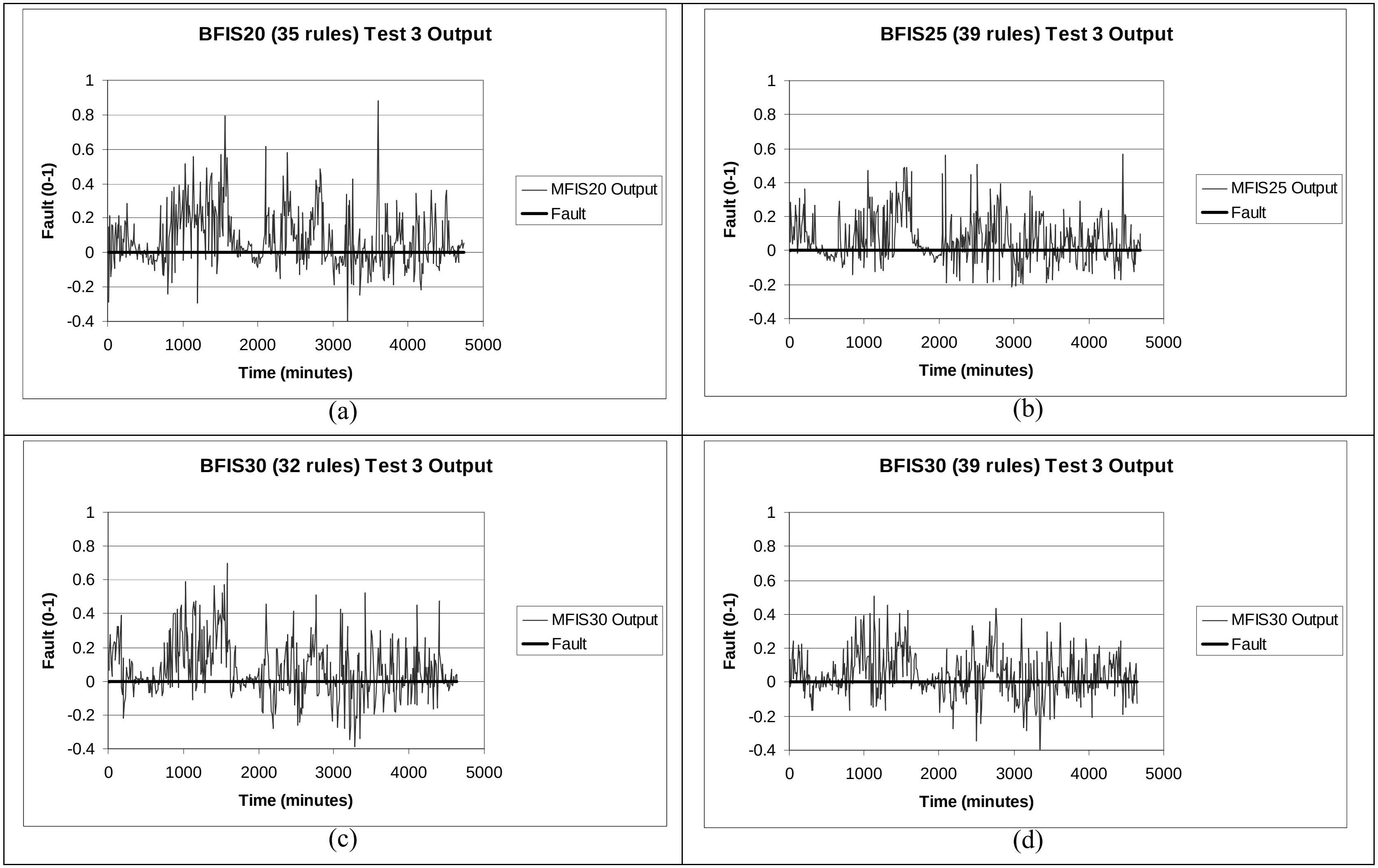

Figure 6.7 Test 3 (Normal Condition) Output for Biological Fault Detection Systems with 10-Minute Interval

The normal test responses are shown in Figure 6.7. All FDI systems had noisy responses. Except for the FDI system with 24 inputs and 35 rules, the FDI systems had no misclassification. The summary of FDI systems performance of test 3 can be seen in Table 6.10.

Table 6.10 FDI Systems Performances of Test 3 (Normal Condition)

FDI Systems	Detection Time	Misclassification	Correct Classification
24 inputs and 35 rules	NA	0.60%	99.40%
29 inputs and 39 rules	NA	0%	100%
34 inputs and 32 rules	NA	0%	100%
34 inputs and 39 rules	NA	0%	100%

6.7 Filter and System Performance

An algorithm for a simple filter is discussed below. In addition to the filter, the algorithm also gives an output of 0 and 1, giving a non-fuzzy output that determines whether the system has a fault or not. This output is important because a person who can read the original graphic might not be around and a simple alarm can be read or even connected to a loudspeaker to be heard by anybody. Below is the step-by-step algorithm to filter the output and give an alarm:

1. Wait until sufficient data sets are available in order for the system to work. If the FDI system uses the prevous 5 sampling steps, the algorithm waits for at least five time steps before giving any output. While waiting the output is zero or no fault.
2. When the output is larger than 1, then it is equal to 1.
3. If the output is lower than 0 then it is equal to 0.
4. When the state of the output changes, observe the next four outputs.

5. If three out of five consecutive output states have changed then the state of the output has changed, otherwise the change of the state is noise. In this case, change the output value to the average of before and after output values.

This filter and fault decision algorithm were tested on test 1 responses of the two best FDI systems of 10-minute interval. The result for the second best system with 29 inputs and 39 rules can be seen on Figure 6.8.

- Figure 6.8a is the original test result for this particular system.
- Figure 6.8b has the filter algorithm output decision of whether the fault has happened.
- Figure 6.8c is the filtered output so that all the noises detected by the filter are removed from the response.
- Figure 6.8d is the noise chart. It shows the points defined as noise in the response by the filter.

The filter performed very well for this FDI system. Figure 6.8b shows that the fault decision exactly follows the real faulty condition of the data set for testing. It can identify the changing condition at step 25, one step after the start of the fault. It also can identify the noise in step 26 that drops the response to 0.19 and marks it as a noise and changes the value to the average of its neighboring points. It also can identify five more points correctly as noise.

The result of the filter and fault decision algorithm tested on the system with 34 inputs and 39 rules can be seen in Figure 6.9. The algorithm classifies points during the noisy transition period from normal to faulty conditions as noise. The fault was introduced in step 20, but the filter algorithm identifies it at step 24 as shown in Figure 6.9b

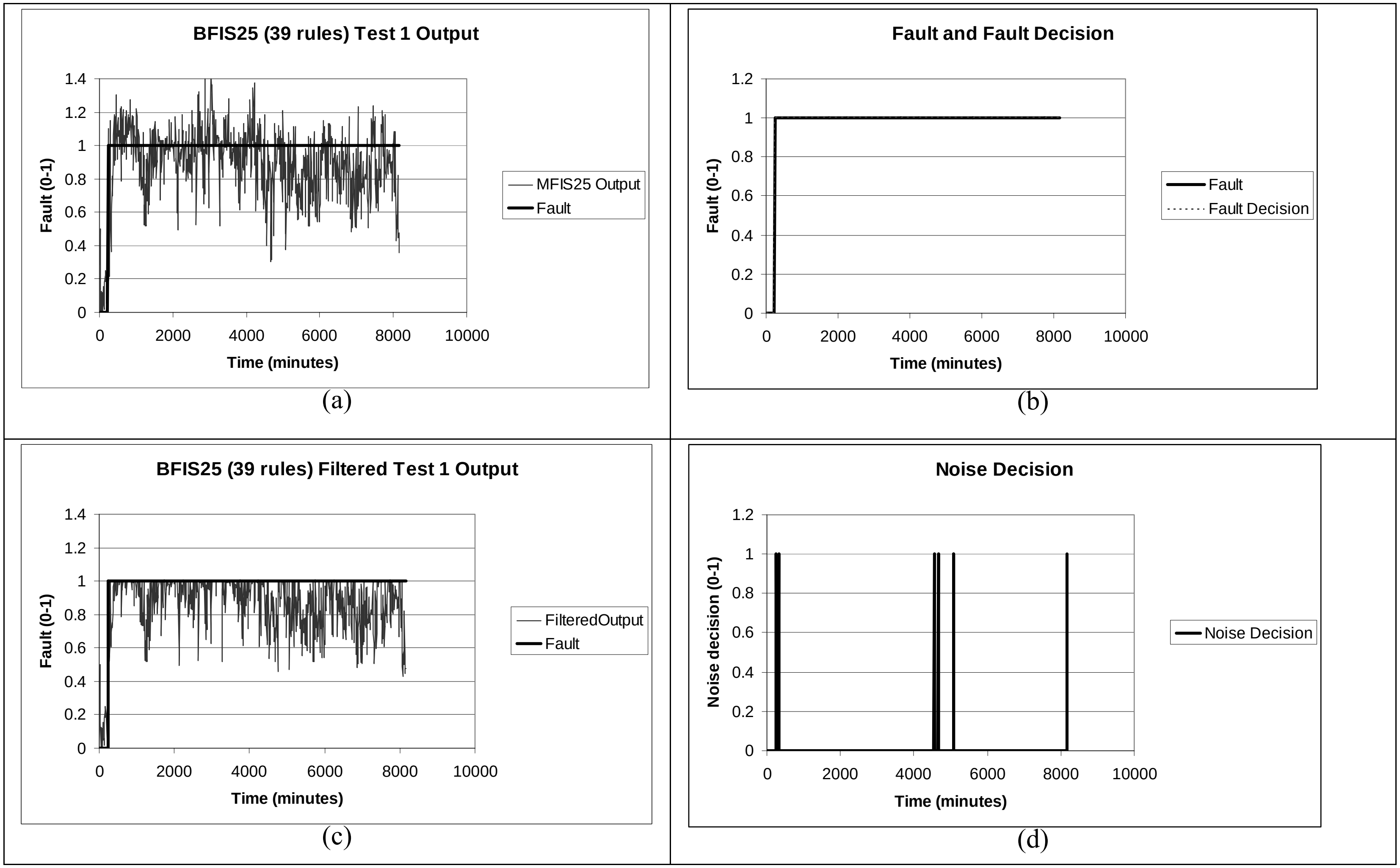

Figure 6.8 Filtered Test 1 Output for BFIS25 (39 Rules)

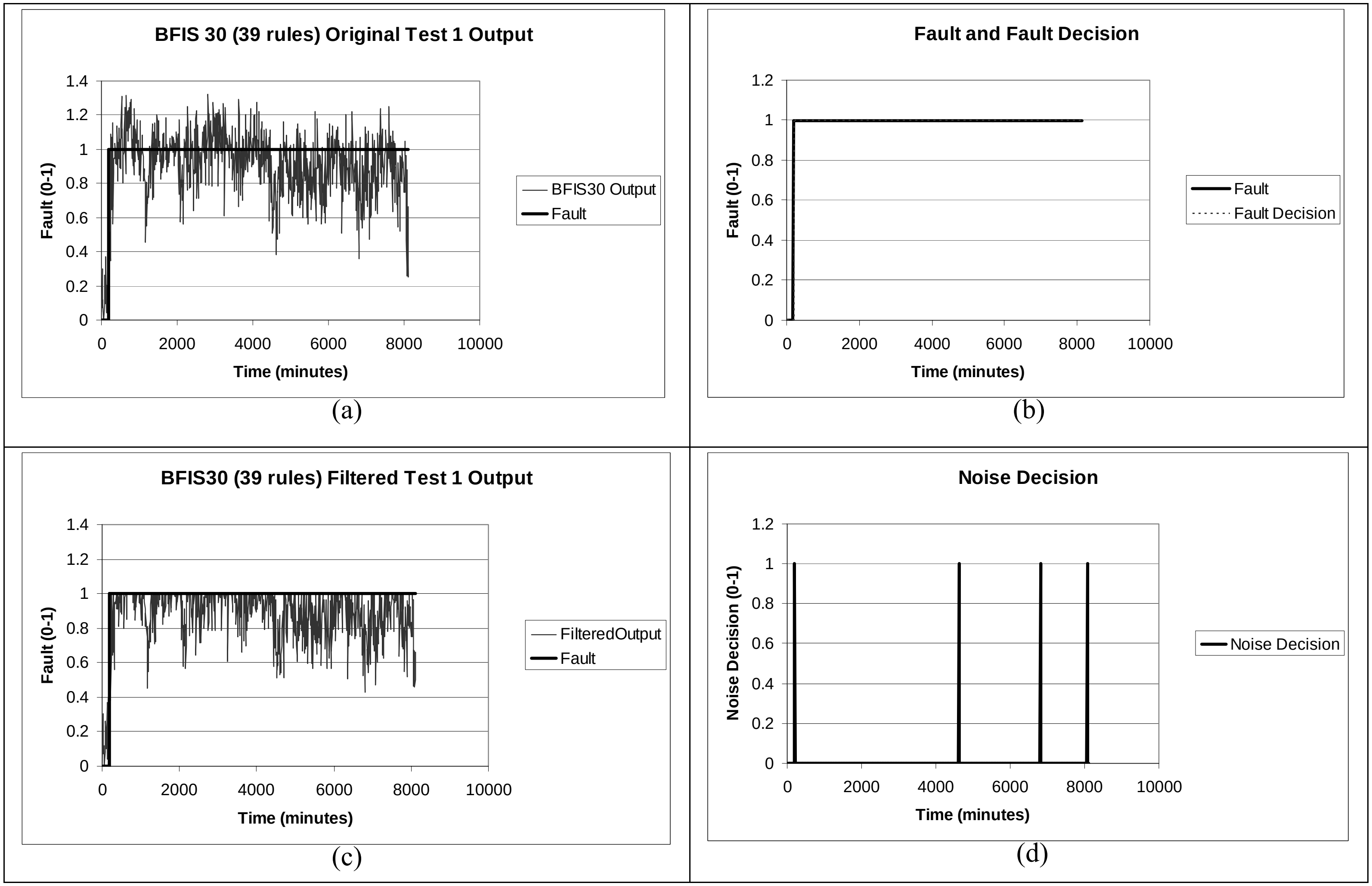

Figure 6.9 Filtered Test 1 Output for BFIS30 (39 rules)

The FDI system response has already changed its state to the faulty condition on step 20 but dropped back to the normal condition for the next three steps. Only after step 24, are there 3 out of 5 consecutive points where the state is in the faulty condition and recognized quickly by the filter algorithm. The detection time of the faulty condition is 30 minutes from the start of the fault. This performance is still very good for a slowly happening transpiration fault and an improvement from 50-minute detection time without the filter.

The filter recognizes four noise points shown by Figure 6.9d. The first noise is the condition at the transition period as described in the paragraph above. Noises in the responses are averaged with their neighboring points as shown in Figure 6.9c. The summary of the FDI system responses with and without the filter is shown in Table 6.11.

Table 6.11 FDI Systems Performances of Test 1 With and Without The Filter

FDI Systems	Detection Time	Misclassification	Correct Classification
34 inputs and 32 rules	50 minutes	2%	98%
34 inputs and 32 rules with filter	10 minutes	0%	100%
34 inputs and 39 rules	50 minutes	0.60%	99.40%
34 inputs and 39 rules with filter	30 minutes	0%	100%

6.8 Result Summary

Table 6.12 FDI Systems Performances of Test 1 (Transpiration Fault)

FDI Systems	*Detection Time*	*Misclassification*	*Correct Classification*
24 inputs and 35 rules	*50 minutes*	*1.80%*	*98.20%*
29 inputs and 39 rules	*50 minutes*	*1%*	*99%*
34 inputs and 32 rules	*50 minutes*	*2%*	*98%*
34 inputs and 39 rules	*50 minutes*	*0.60%*	*99.40%*

Table 6.12 shows the NF FDI system performances in the processing test 1 data set. The responses are noisy which means this particular pattern is weakly

recognized within the system's noise. This condition caused long detection times. The correct classification percentage is at or above 99% for the two best responses.

Table 6.13 FDI Systems Performances of Test 1 With and Without Filter

FDI Systems	*Detection Time*	*Misclassification*	*Correct Classification*
34 inputs and 32 rules	*50 minutes*	*2%*	*98%*
34 inputs and 32 rules with filter	*10 minutes*	*0%*	*100%*
34 inputs and 39 rules	*50 minutes*	*0.60%*	*99.40%*
34 inputs and 39 rules with filter	*30 minutes*	*0%*	*100%*

Filtering helps reduces both the detection time and misclassification in the noisy system responses as shown in Table 6.13. The misclassification for the noisy FDI system responses decreasedfrom 2% and 0.6% to zero for the two best systems.

Table 6.14 FDI Systems Performances of Test 2 (Transpiration Fault)

FDI Systems	*Detection Time*	*Misclassification*	*Correct Classification*
24 inputs and 35 rules	*10 minutes*	*0.90%*	*99.10%*
29 inputs and 39 rules	*10 minutes*	*0%*	*100%*
34 inputs and 32 rules	*10 minutes*	*0%*	*100%*
34 inputs and 39 rules	*10 minutes*	*0%*	*100%*

Test 2 responses have very little noise, signaling this particular pattern is strongly recognized by all FDI systems. The FDI systems performances in the processing test 2 data set can be seen in Table 6.14. All systems recognized the fault early and misclassification percentage is 0% for three out of four systems.

Table 6.15 FDI Systems Performances of Test 3 (Normal Condition)

FDI systems	*Detection time*	*Misclassification*	*Correct classification*
24 inputs and 35 rules	*NA*	*0.60%*	*99.40%*
29 inputs and 39 rules	*NA*	*0%*	*100%*
34 inputs and 32 rules	*NA*	*0%*	*100%*
34 inputs and 39 rules	*NA*	*0%*	*100%*

The normal conditions can successfully be recognized by the FDI systems. Three out of four systems have 100% correct classification. The NF biological FDI system successfully identified the transpiration fault in the hydroponic plant production system. The average detection time of 30 minutes is fast enough for early fault detection.

CHAPTER 7

NEURO-FUZZY MECHANICAL FAULT DETECTION AND IDENTIFICATION SYSTEM

7.1 Mechanical Fault in The Hydroponic System.

Mechanical faults in the hydroponic system can be categorized into abrupt faults and incipient faults. Two different types of experiments from each category were performed from November 2000 to June 2001. Malfunctioning episodes of the pH control pump and the hydroponic system circulation were performed to simulate abrupt faults. Drifting of the pH sensor and EC sensor were performed to simulate incipient faults.

The pH control pump was deemed to be the most important fault. If the pump that supplied acid stayed on without any request from control signal for a long period of time, the pH would drop quickly to a level that could destroy the plants in less than an hour. Since doing this exact experiment could destroy all of the plants very quickly, the opposite fault by stopping the pump was performed instead. If the act of stopping the pump could be detected quickly, the fault of continuously on could be detected as quickly.

The second important fault was the sudden stopping of the circulation pump. Without water circulation, feedback from the sensor could be delayed. In case of the DO control, the DO sensor reading stayed low even if the DO had been fully supplied for a while. This had no negative effect on the plants. This was a different matter from the pH control. If the pH rose above 5.8, the pH pump tried to supply more and more acid to the circulation system. But, the acid did not get to the sensor for a long time and finally acidified the solution excessively. The sensor was slow to recognize this condition which caused the plants to die.

The third important fault was a slowly drifting pH sensor. Undetected sensor drift caused the pH of the nutrient solution move outside the optimal range. This caused the plant to absorb less nutrient and had slower growth. This situation reduced the amount of plant production and the quality of the product.

The slowly drifting EC sensor caused a similar problem as the pH sensor although it was not significant. The value of the nutrient concentration that was lower than the optimal range decreased the amount of nutrient available for absorption and the higher value could hinder the nutrient absorption by creating more osmosis barrier. This also caused the plant to absorb less nutrient and leading to slower growth.

7.2 Neuro-Fuzzy FDI Specifications

Neuro-fuzzy systems with 14, 19, 24, 29 and 34 variables were used as FDI systems. Systems with 14 inputs were developed only for 10-minute interval data since the number of inputs is too low to give satisfactory results.

The inputs used in the FDI systems were: pH, dissolved oxygen (DO), electrical conductivity (EC), pH control signal (pHcs), DO control signal (DOcs), air temperature, light intensity, relative humidity (RH), solution temperature, and previous sample values of pH, DO, EC, pHcs, and DOcs.

Subtractive clustering was used to extract the neuro-fuzzy fault rules from the input-output data. Range of influence (*roi*) coefficients in the clustering method determined how many cluster centers formed. Values between 0.2 and 0.5 are recommended [Chu 1994]. Several *roi* values were used to form the neuro-fuzzy systems. A small *roi* means a short range of influence of the cluster center and a large number of cluster centers formed. The number of cluster center determines the number of fuzzy rules. For example the subtractive clustering algorithm with a roi value of 0.234 formed 17 cluster centers (rules) for the FDI system with 29 inputs.

There was only one output variable, which reduced the complexity of the neuro-fuzzy detection system. In place of a different output for each fault condition, different values of the single output were used. The output value of 1 represented the normal condition. The value of 2 represented the pH control pump fault. The value of 3 represented the circulation pump fault. The value of 4 represented the slowly drifting pH sensor fault and the value of 5 represented the slowly drifting EC sensor fault.

The placement of the faults relative to normal in the multilevel value of the FDI system output was based on how important the fault was to the system. By placing the most important fault next to the normal condition, it had more sensitivity and less noise. It was hypothesized that a desirable fault detection system was the one that performed the best in detecting these important faults and performed at least average in detecting the least important faults in the system.

7.3 Data Sets

Experiment data sets were categorized into training and testing data sets. Training data sets used in forming and training of the neuro-fuzzy systems can be seen in Table 7.1. The testing datasets are shown in Table 7.2. The number of data entries for systems with 5-minute intervals was twice the number of data entries for systems with 10-minute intervals because the same number of experiments was used.

Table 7.1 Training Data Sets

Experiment Type	Tank #	Start Date	Start Time	End Date	End Time
Normal	2	02/22/01	6am	02/24/01	6am
Normal	2	02/24/01	6am	02/26/01	6am
Normal	2	02/26/01	6am	02/27/01	6am
Normal	2	03/06/01	6am	03/08/01	6am
Normal	2	04/20/01	6am	04/22/01	6am
Normal	2	04/25/01	6am	04/26/01	6am
Normal	2	05/14/01	6am	05/16/01	6am
pHp[a]	2	11/08/00	3:30pm	11/10/00	12:35pm
pHp	2	11/12/00	2:15pm	11/14/00	1:45pm
pHp	3	02/22/01	11:55am	02/24/01	12pm
pHp	3	02/26/01	12:55pm	02/27/01	5:55pm
pHp	3	02/28/01	10:10am	03/02/01	4:45pm
pHp	3	05/25/01	12:28pm	05/26/01	6:20pm
pHp	3	05/31/01	12:20p m	06/01/01	5:45pm
CP[b]	3	11/18/00	1:30pm	11/20/00	1:45pm
CP	1	12/07/00	6:01am	12/07/00	1:15pm
CP	1	12/13/00	7:11am	12/13/00	1:41pm
CP	3	03/03/01	11:53am	03/03/01	4:50pm
CP	3	03/06/01	12:30pm	03/06/01	5:47pm
CP	3	03/08/01	1:10pm	03/08/01	5:40pm
CP	3	03/09/01	12:06pm	03/09/01	4:30pm
CP	3	03/15/01	12:05pm	03/15/01	6:20pm
CP	3	04/16/01	11:58am	04/16/01	4:45pm
CP	3	04/18/01	11:59am	04/18/01	5:30pm
CP	3	05/16/01	12:06pm	05/16/01	5:45pm
CP	3	05/18/01	12:01pm	05/18/01	5:30pm
CP	3	05/22/01	1:07pm	05/22/01	5:35pm
CP	3	05/24/01	12:16pm	05/24/01	6pm
pHs[c]	1	05/09/01	12:39pm	05/10/01	4:32pm
pHs	1	05/11/01	12pm	05/12/01	4pm
pHs	1	05/17/01	12:05pm	05/18/01	5:30pm
pHs	1	05/21/01	1:18pm	05/22/01	5:35pm
pHs	1	05/29/01	12:01pm	05/30/01	6:30pm
ECs[d]	1	04/11/01	12:58pm	04/13/01	1:08pm
ECs	1	05/01/01	12:10pm	05/03/01	1:01pm
ECs	1	05/03/01	1:28pm	05/07/01	12:12pm
ECs	1	05/07/01	12:35pm	05/09/01	12pm

[a]pHp : pH control pump
[b]CP : circulation pump
[c]pHs : pH sensor
[d]ECs : electrical conductivity sensor

Table 7.2 Testing Data Sets

Experiment Type	Tank #	Start Date	Start Time	End Date	End Time
Normal	2	01/14/01	6am	01/15/01	6am
Normal	2	03/26/01	6am	03/27/01	6am
pHp[a]	3	02/20/01	12:25pm	02/21/01	5:45pm
pHp	3	05/29/01	12:01pm	05/30/01	6:30pm
CP[b]	1	12/05/00	1:44pm	12/05/00	5:35pm
CP	3	04/20/01	11:54am	04/20/01	5:15pm
pHs[c]	1	11/29/00	12:03pm	11/30/01	4pm
pHs	1	05/15/01	12:20pm	05/16/01	5:45pm
ECs[d]	1	12/10/00	10:54am	12/12/00	7:12am
ECs	1	04/30/01	12:20pm	05/01/01	12pm

[a]pHp : pH control pump
[b]CP : circulation pump
[c]pHs : pH sensor
[d]ECs : electrical conductivity sensor

7.4 Training Results

The FDI systems constructed by the subtractive clustering method were trained further using training data from the fault experiments. Each system was trained for 5 epochs with each epoch consisting of 500 iterations (a total of 2500 iterations). The training result for Mechanical Fault Detection System with 5-minute interval (FDI5 systems) is shown in Table 7.3.

The starting errors of the newly constructed fault detection systems from the subtractive clustering method were fairly large but quickly diminished starting with the first training. The first epoch of the training reduced the error the most while the last epoch reduced it the least. The number of training epochs was limited to 5, since further training did not give significant improvement. All systems were trained to the same number of epochs to have the same state of training for comparison.

Table 7.3 Training Result of Mechanical Fault Detection Systems with 5-Minute Interval

		The 1st Epoch		The 2nd Epoch		The 3rd Epoch		The 4th Epoch		The 5th Epoch	
Roi	**Rules**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**
19 INPUTS											
0.25	14	0.7974	0.5757	0.5757	0.5668	0.5668	0.5639	0.5639	0.5618	0.5618	0.561
0.3	8	0.8573	0.6187	0.6187	0.6106	0.6106	0.6067	0.6067	0.6045	0.6045	0.6035
24 INPUTS											
0.22	18	0.7459	**0.451**[a]	0.451	**0.441**[a]	0.441	**0.4371**[a]	0.4371	**0.4336**[a]	0.4336	**0.4318**[a]
0.25	13	0.7793	0.5144	0.5144	0.4855	0.4855	0.4733	0.4733	**0.4633**[d]	0.4633	**0.4539**[c]
0.28	9	0.8217	0.5567	0.5567	0.5115	0.5115	0.5048	0.5048	0.5024	0.5024	0.5001
0.3	8	0.8406	0.5836	0.5836	0.5346	0.5346	0.5248	0.5248	0.5192	0.5192	0.5166
29 INPUTS											
0.234	17	0.7198	**0.4708**[b]	0.4708	**0.4587**[b]	0.4587	**0.4551**[b]	0.4551	**0.4522**[b]	0.4522	**0.4509**[b]
0.24	16	0.7216	**0.4755**[c]	0.4755	**0.4598**[c]	0.4598	**0.4576**[c]	0.4576	**0.4558**[c]	0.4558	**0.4542**[d]
0.25	13	0.7663	0.4833	0.4833	0.4761	0.4761	0.4734	0.4734	0.4715	0.4715	0.4699
34 INPUTS											
0.26	11	0.793	0.5348	0.5348	0.495	0.495	0.4793	0.4793	0.4749	0.4749	0.4731
0.28	8	0.8542	**0.4822**[d]	0.4822	**0.4746**[d]	0.4746	**0.471**[d]	0.471	0.4685	0.4685	0.4672
0.3	7	0.8549	0.6155	0.6155	0.565	0.565	0.5379	0.5379	0.5281	0.5281	0.5279

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

Table 7.4 Training Results for Mechanical Fault Detection Systems with 10-Minute Interval.

		The 1st epoch		The 2nd epoch		The 3rd epoch		The 4th epoch		The 5th epoch	
roi	**rules**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**	**Start Error**	**End Error**
						14 INPUTS					
0.25	13	0.8271	0.5693	0.5693	0.5628	0.5628	0.5602	0.5602	0.5593	0.5593	0.5586
0.3	7	0.8718	0.6381	0.6381	0.6408	0.6381	0.6332	0.6332	0.6304	0.6304	0.6275
						19 INPUTS					
0.22	17	0.7693	0.5626	0.5626	0.5578	0.5578	0.5553	0.5553	0.5522	0.5522	0.5515
0.25	13	0.8083	0.5625	0.5625	0.5511	0.5511	0.5467	0.5467	0.5442	0.5442	0.5423
0.27	11	0.8164	0.5921	0.5921	0.583	0.583	0.5803	0.5803	0.5793	0.5793	0.5783
0.3	7	0.864	0.6096	0.6096	0.6057	0.6057	0.6028	0.6028	0.6002	0.6002	0.5985
						24 INPUTS					
0.23	17	0.7646	**0.5327** [b]	0.5327	**0.5259** [b]	0.5259	**0.5235** [b]	0.5235	**0.5215** [b]	0.5215	**0.5195** [c]
0.25	13	0.7096	0.5526	0.5526	**0.5454** [d]	0.5454	0.5418	0.5418	0.5394	0.5394	0.5375
0.27	11	0.81	0.5543	0.5543	0.548	0.548	0.5458	0.5458	0.5443	0.5443	0.543
0.3	8	0.8526	0.5945	0.5945	0.5894	0.5894	0.5877	0.5877	0.5867	0.5867	0.5859
						29 INPUTS					
0.24	16	0.7568	**0.5199** [a]	0.5199	**0.5143** [a]	0.5143	**0.5116** [a]	0.5116	**0.5103** [a]	0.5103	**0.5092** [a]
0.25	13	0.7873	**0.5523** [d]	0.5523	**0.5435** [c]	0.5435	**0.5415** [d]	0.5415	0.5415	0.5406	0.5399
0.27	11	0.83113	0.5632	0.5632	0.5552	0.5552	0.5492	0.5492	0.5445	0.5445	0.5373
0.3	8	0.8496	0.5781	0.5781	0.5702	0.5702	0.566	0.566	0.5636	0.5636	0.5611
						34 INPUTS					
0.26	11	0.7875	0.5568	0.5568	0.5477	0.5477	0.5424	0.5424	**0.5382** [d]	0.5382	**0.5354** [d]
0.28	10	0.8047	**0.5459** [c]	0.5459	0.5359	0.5359	**0.5293** [c]	0.5293	**0.5236** [c]	0.5236	**0.5191** [b]
0.3	8	0.8411	0.5734	0.5734	0.5592	0.5592	0.5536	0.5536	0.5497	0.5497	0.5474

[a] the best training result [b] the 2^{nd} best [c] the 3^{rd} best [d] the 4^{th} best

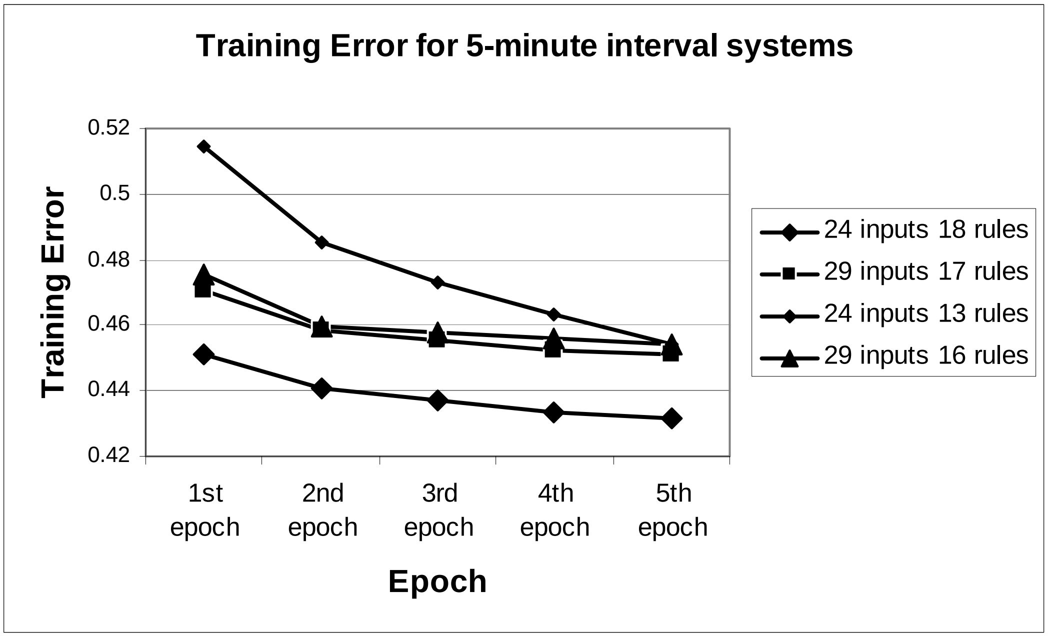

Figure 7.1 Five-Minute Interval Systems' Error Trend

The five-minute interval FDI system error trends for every epoch of the training can be seen in Figure 7.1. The last epoch training result of the FDI systems are very close to each other, except for the best system. Systems with more rules had smaller training errors than those with fewer rules.

The system with 34 inputs had more training error than 24 and 29 input systems. Generally the more input available the better the system capability for approximating the monitored system. Systems with more inputs need more training data sets. In this case, systems with 34 inputs were limited to have a maximum number of 11 rules. As a comparison, the system with 29 inputs had a maximum number of 17 rules.

Table 7.4 shows the training result for Mechanical FDI systems with 10-minute intervals (FDI10 systems). The final training errors for these systems are generally larger than the 5-minute interval (FDI5) systems. The system with 29 inputs and 16 rules had the least error.

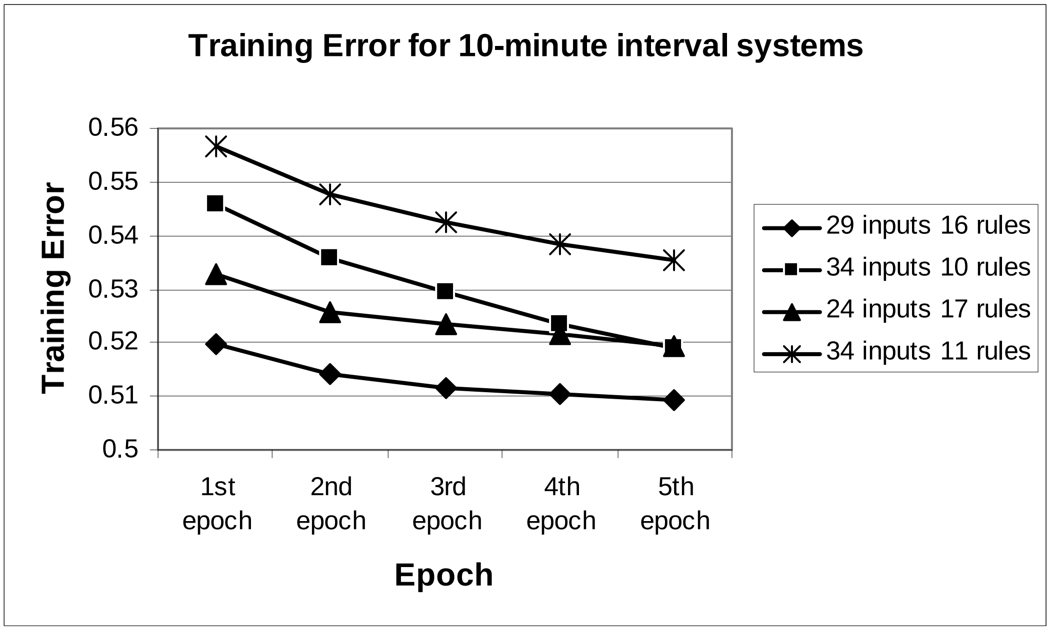

Figure 7.2 Ten-Minute Interval Systems' Error Trend

The 10-minute interval error trend is shown in Figure 7.2 for every epoch of training. All errors are uniformly separated at the beginning of the training. At the end of the training, the system with 34 inputs and 10 rules and the system with 24 inputs and 17 rules had similar error values. In this case more inputs gave more degrees of freedom in modeling the process than more rules. Systems with 34 inputs had fewer rules and were limited by the available training data sets but had training errors comparable to 29 input systems.

7.5 Sensitivity Test

The sensitivity of the neuro-fuzzy system to its inputs was explored. In order to do this, one of the input variables was removed at a time from the training data set. The ANFIS was trained with this data set for 500 iterations. The result can be seen in Table 7.5.

Table 7.5 Training Error of 5-Minute Interval ANFIS
with One Variable Eliminated at a Time for 24 Inputs and 18 Rules

Variable Removed	Training Error
Electrical Conductivity	0.8107
Relative Humidity	0.7545
Light Intensity	0.7199
Nutrient Temperature	0.7159
Air Temperature	0.7073
pH control signal	0.6982
Dissolved Oxygen	0.6854
pH	0.6583
Dissolved Oxygen control signal	0.6406

The elimination of the electrical conductivity caused the largest training error of 0.8107. The next ones were the relative humidity, the light intensity, the nutrient temperature and the air temperature. So it can be deduced that the aerial variables are more important than the nutrient solution variables in the neuro-fuzzy fault detection system. The aerial variables are needed to determine the effect of seasonal changes and also they are important for the adaptivity of the FDI system in different weather conditions. The high sensitivity of the neuro-fuzzy system to the EC input is an exception. EC was maintained manually and there was no additional control signal as an input variable. So the EC input is the only variable available for the EC sensor fault detection. The elimination of the input variables used in the neuro-fuzzy FDI system caused significant increase in the training error. The least error increase caused by the dissolved oxygen control signal elimination is 0.6406, much larger than 0.451 when all inputs are available.

7.6 FDI System Performance Definitions

The FDI system performance is measured from detection time and correct classification of faults. The detection time is the time needed by the FDI system to detect the occurrence of fault in the monitored system.

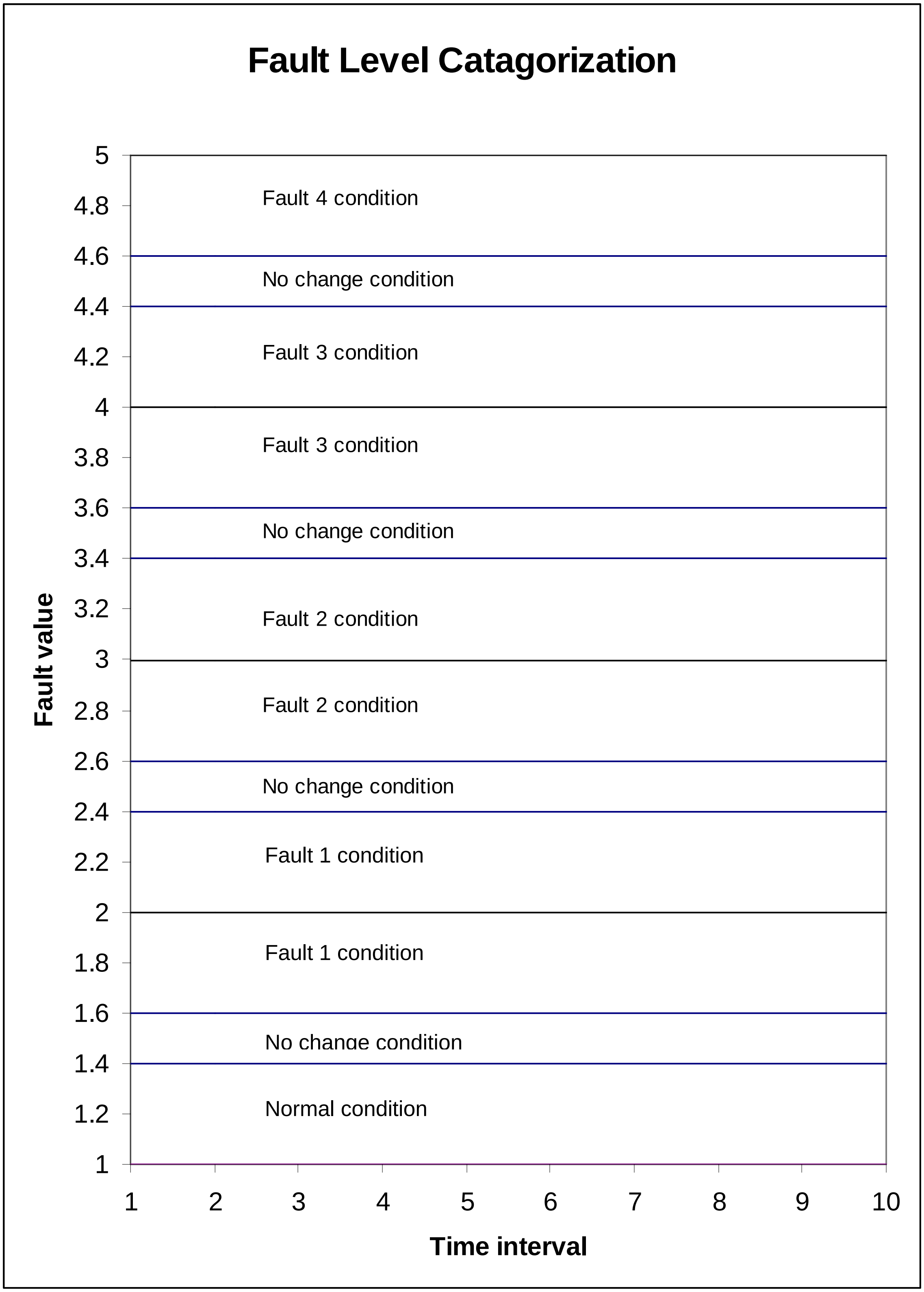

Figure 7.3 Fault Level Categorization

The FDI system makes a correct classification if the system output shows the correct level for the intended fault after the fault detection. Misclassification does not include discrepancies at the output during the detection time. The fault level categorization is shown in Figure 7.3.

The area between 1 and 2 can be divided into three parts. The value between 1 and 1.4 is defined as normal, between 1.4 and 1.6 is defined as no change from the previous condition, and between 1.6 and 2 is defined as fault 1 condition. For example at time $t-1$ (one sampling step before the current sampling instant), assume the response y is 1.7, representing fault 1 condition. At time t, $y=1.5$ and according to this definition, the condition at time t is fault 1. If at time $t+1$, $y= 1.35$ then the response has changed to normal condition. The same definition is applicable for the area between fault value 2 and 3, fault value 3 and 4 and so on.

7.7 Testing Results

To make sure that all the systems were not over trained, some data sets were used for testing the trained FDI systems. The results for 5-minute interval (FDI5) and 10-minute interval (FDI10) systems are shown in Table 7.6 and Table 7.7. There were 5 different conditions for the mechanical FDI systems so there were 5 different tests for each condition: test 1 for the normal condition, test 2 for the pH pump faulty condition, test 3 for the circulation pump faulty condition, test 4 for the pH sensor faulty condition, and test 5 for the EC sensor faulty condition.

From the testing results in Table 7.6, the four best FDI5 systems are:

1. 19 input system with 14 rules.
2. 29 input system with 17 rules.
3. 24 input system with 18 rules.
4. 24 input system with 13 rules.

Table 7.6 Testing Result of Mechanical Fault Detection System with 5-Minute Interval

Roi	Rules	Training Error	Testing Error: Test 1	Test 2	Test 3	Test 4	Test 5	Error Sum
					19 INPUTS			
0.25	14	0.561	0.16928524	**0.479584081** [c]	**0.488848056**[b]	**0.632452719** [a]	0.769139544	**2.53930964** [a]
0.3	8	0.6035	0.172303441	0.581795916	0.665454077	0.925005074	1.000119722	3.344678231
					24 INPUTS			
0.22	18	**0.4318** [a]	0.172978905	0.524922662	0.539456798	**0.74286112** [c]	**0.701994717** [b]	**2.682214202** [c]
0.25	13	**0.4539** [c]	0.154150573	0.565303998	0.549193708	**0.732092524** [b]	**0.716150722** [c]	**2.716891524** [d]
0.28	9	0.5001	0.14893771	0.519931468	0.542731378	1.122756089	0.994240583	3.328597229
0.3	8	0.5166	0.148875552	0.5280686	0.64655073	1.225181895	0.880655581	3.429332359
					29 INPUTS			
0.234	17	**0.4509** [b]	**0.129831308** [a]	**0.474498701** [b]	0.581590867	**0.757940106** [d]	**0.726103723** [d]	**2.669964704** [b]
0.24	16	**0.4542** [d]	**0.134174631** [b]	**0.463846389** [a]	**0.536137192** [d]	0.861009851	0.856645327	2.85181339
0.25	13	0.4699	0.181670767	**0.519768732** [d]	**0.345700133** [a]	1.159686144	0.804532003	3.011357778
					34 INPUTS			
0.26	11	0.4731	0.212697216	0.576912647	0.646172883	0.817985037	**0.644045139** [a]	2.897812922
0.28	8	0.4672	**0.148063709** [d]	0.559363674	**0.491292428** [c]	0.911436059	0.915560502	3.025716371
0.3	7	0.5279	**0.14473702** [c]	0.531180859	0.849093149	1.176408171	1.17664346	3.878062659

[a] the best training result [b] the 2^{nd} best [c] the 3^{rd} best [d] the 4^{th} best

Table 7.7 Testing result of Mechanical Fault Detection System with 10-Minute Interval

Roi	Rules	Training Error	Testing Error: Test 1	Test 2	Test 3	Test 4	Test 5	Error Sum
14 INPUTS								
0.25	13	0.5586	0.170825983	0.499935284	0.469445776	**0.697353449** [c]	1.011311096	2.848871589
0.3	7	0.6275	0.170704827	**0.477407018** [d]	0.90162597	1.388486755	1.446180149	4.384404719
19 INPUTS								
0.22	17	0.5515	**0.118790906** [a]	**0.456828864** [b]	0.522521235	1.090957667	**0.858216506** [c]	3.047315178
0.25	13	0.5423	0.170436178	0.550154877	0.631541893	0.765149135	0.885104496	3.002386578
0.27	11	0.5783	0.184652696	0.524974223	0.521585261	0.967780144	0.964602487	3.163594812
0.3	7	0.5985	0.145250984	0.580298248	0.566414791	1.431709687	1.172286923	3.895960633
24 INPUTS								
0.23	17	**0.5195** [c]	0.180146431	0.52243554	**0.416614223** [c]	**0.587299059** [a]	0.887727578	***2.594222831*** [b]
0.25	13	0.5375	0.213459263	0.500128201	**0.332069123** [a]	**0.643849741** [b]	0.886023789	**2.575530118** [a]
0.27	11	0.543	0.220109543	0.519935463	0.519077832	0.835028521	**0.868539768** [d]	2.962691127
0.3	8	0.5859	**0.140891892** [d]	0.568474568	0.565132209	1.060458028	0.933372578	3.268329276
29 INPUTS								
0.24	16	**0.5092** [a]	0.183878419	**0.467728577** [c]	0.522634642	0.749475577	**0.790078102** [a]	**2.713795317** [d]
0.25	13	0.5399	0.22214617	**0.445758251** [a]	0.525904196	0.846688201	**0.850907359** [b]	2.891404177
0.27	11	0.5373	0.17889548	0.522971074	0.508016431	0.824622695	1.017390799	3.051896479
0.3	8	0.5611	0.143157121	0.510632408	0.632110595	1.331521721	0.957864084	3.575285929
34 INPUTS								
0.26	11	**0.5354** [d]	**0.138879251** [b]	0.555999751	**0.419449975** [d]	0.831724637	1.175339415	3.121393028
0.28	10	**0.5191** [b]	0.176137416	0.531102764	**0.373036003** [b]	**0.724460264** [d]	0.902113771	**2.706850218** [c]
0.3	8	0.5474	**0.13987442** [c]	0.525451429	0.579274031	1.000140976	1.000053852	3.244794708

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

From the testing results in Table 7.7, the four best FDI10 systems are:

1. 24 input system with 13 rules
2. 24 input system with 17 rules
3. 34 input system with 10 rules.
4. 29 input system with 16 rules.

Systems with the least testing error sums for both the FDI5 and the FDI10 systems are:

1. The best is from the 5-minute interval.
2. The second best from the 10-minute interval.
3. The third best is from the 10-minute interval and
4. The fourth best is from the 5-minute interval.

The best individual test results were also expected to alternate between the FDI5 and the FDI10 systems, but the FDI10 systems had the best test result from the test number 1 to 4. The FDI5 systems had the best response for only the test number five. The overall test results are similar because the error for the test number 5 is much larger for the FDI10 systems than for the FDI5 systems so the final errors are balanced out. This result is unexpected since the training results for the FDI5 systems are much better than for FDI10 systems. To see whether the FDI5 is over trained, the two best-trained FDI5 systems were tested for every epoch of training and shown in Table 7.8. As comparison, two best FDI10 systems were also tested and the results are shown in Table 7.9.

Table 7.8 Two Best-Trained 5-Minute Interval System Testing Result for Each Training Epoch

Epoch #	Test 1	Test 2	Test 3	Test 4	Test 5	Error Sum
			Five Minute Interval System with roi=0.22, 24 inputs and 18 rules (Training Error=0.4318)			
1	**0.143544213** [a]	**0.524400917** [a]	**0.53798455** [b]	0.780729701	0.746990416	2.733649797
2	0.173386371	**0.528251454** [d]	**0.53219394** [a]	**0.754967277** [d]	**0.708019107** [d]	**2.69681815** [d]
3	**0.172978905** [d]	**0.524922662** [b]	**0.539456798** [c]	**0.74286112** [c]	**0.701994717** [c]	**2.682214202** [c]
4	**0.16996233** [c]	**0.52800073** [c]	**0.544957508** [d]	**0.717618945** [b]	**0.69431375** [b]	**2.654853262** [b]
5	**0.169661955** [b]	0.528259882	0.546912877	**0.707092742** [a]	**0.685374525** [a]	**2.637301982** [a]
			Five Minute Interval System with roi=0.234, 29 inputs and 17 rules (Training Error=0.4509)			
1	0.135287767	**0.463247468** [a]	**0.560780431** [a]	0.87154058	0.79254761	2.823403857
2	**0.131713244** [d]	**0.474586913** [c]	**0.578098437** [b]	**0.781487984** [d]	**0.74645471** [d]	**2.712341288** [d]
3	**0.130549678** [c]	**0.475159238** [d]	**0.57905351** [d]	**0.766294653** [c]	**0.731671745** [c]	**2.682728823** [c]
4	**0.130389924** [b]	0.47588489	**0.57845429** [c]	**0.762273999** [b]	**0.724728412** [a]	**2.671731516** [b]
5	**0.129831308** [a]	**0.474498701** [b]	0.581590867	**0.757940106** [a]	**0.726103723** [b]	**2.669964704** [a]

[a] the best training result [b] the 2nd best [c] the 3rd best [d] the 4th best

Table 7.9 Two Best-Trained 10-Minute Interval System Testing Result for Each Training Epoch

Epoch #	Test 1	Test 2	Test 3	Test 4	Test 5	Error Sum
			Ten Minute Interval System with roi=0.24, 29 inputs and 16 rules (Training Error=0.5092)			
1	**0.183865569 [b]**	0.483745469	0.546850185	0.768141602	0.870364086	2.85296691
2	**0.183419626 [a]**	**0.475505746 [d]**	**0.530374222 [d]**	**0.756841708 [d]**	**0.842872643 [d]**	**2.789013944 [d]**
3	0.184364957	**0.470459135 [c]**	**0.526809382 [c]**	**0.749742356 [b]**	**0.817077319 [c]**	**2.748453149 [c]**
4	**0.184117254 [d]**	**0.469377615 [b]**	**0.523582262 [b]**	**0.749892153 [c]**	**0.802072993 [b]**	**2.729042278 [b]**
5	**0.183878419 [c]**	**0.467728577 [a]**	**0.522634642 [a]**	**0.749475577 [a]**	**0.790078102 [a]**	**2.713795317 [a]**
			Ten Minute Interval System with roi=0.28, 34 inputs and 10 rules (Training Error=0.5191)			
1	0.18424418	**0.540111961 [d]**	0.547545845	0.8557179	0.970288411	3.097908297
2	**0.18306252 [d]**	**0.538696929 [c]**	**0.508060257 [d]**	**0.81815396 [d]**	**0.948539754 [d]**	**2.99651342 [d]**
3	**0.176856637 [c]**	0.542277791	**0.444994882 [c]**	**0.782580062 [c]**	**0.941813051 [c]**	**2.888522422 [c]**
4	**0.175929834 [a]**	**0.533702131 [b]**	**0.370003695 [a]**	**0.76048469 [b]**	**0.926503068 [b]**	**2.766623418 [b]**
5	**0.176137416 [b]**	**0.531102764 [a]**	**0.373036003 [b]**	**0.724460264 [a]**	**0.902113771 [a]**	**2.706850218 [a]**

[a] the best training result [b] the 2^{nd} best [c] the 3^{rd} best [d] the 4^{th} best

The testing results for the mechanical multilevel value FDI systems were not as clear as the biological fault detection system. In the biological fault detection system, all test results were better with additional epochs of training. This was not the case with the multilevel value fault detection where some have the best test results for the first epoch of training and some have the best results at the last epoch of training.

Although the FDI5 systems did not do as well as the FDI10 systems, the testing error sum was still decreasing with each additional epoch of training. The additional error for tests that reached their best result at the first or second epoch of training were small compared with the reduction in error for tests that reached their best result at the last epoch of training. The FDI10 systems had more uniform test result in every epoch with the least errors reached mostly at the last epoch of training. More training reduced the testing errors more than the FDI5 systems.

These test results confirmed that both the FDI5 and the FDI10 systems were not overtrained. The FDI10 systems processed the input patterns of the test files better than the FDI5 systems.

Although testing errors in Table 7.8 and Table 7.9 are good indicators of the FDI systems ability in processing the data sets, they do not show the dynamic of the response. For this purpose, the best four FDI5 and FDI10 system response charts are shown in Figure 7.4 and 7.5 .

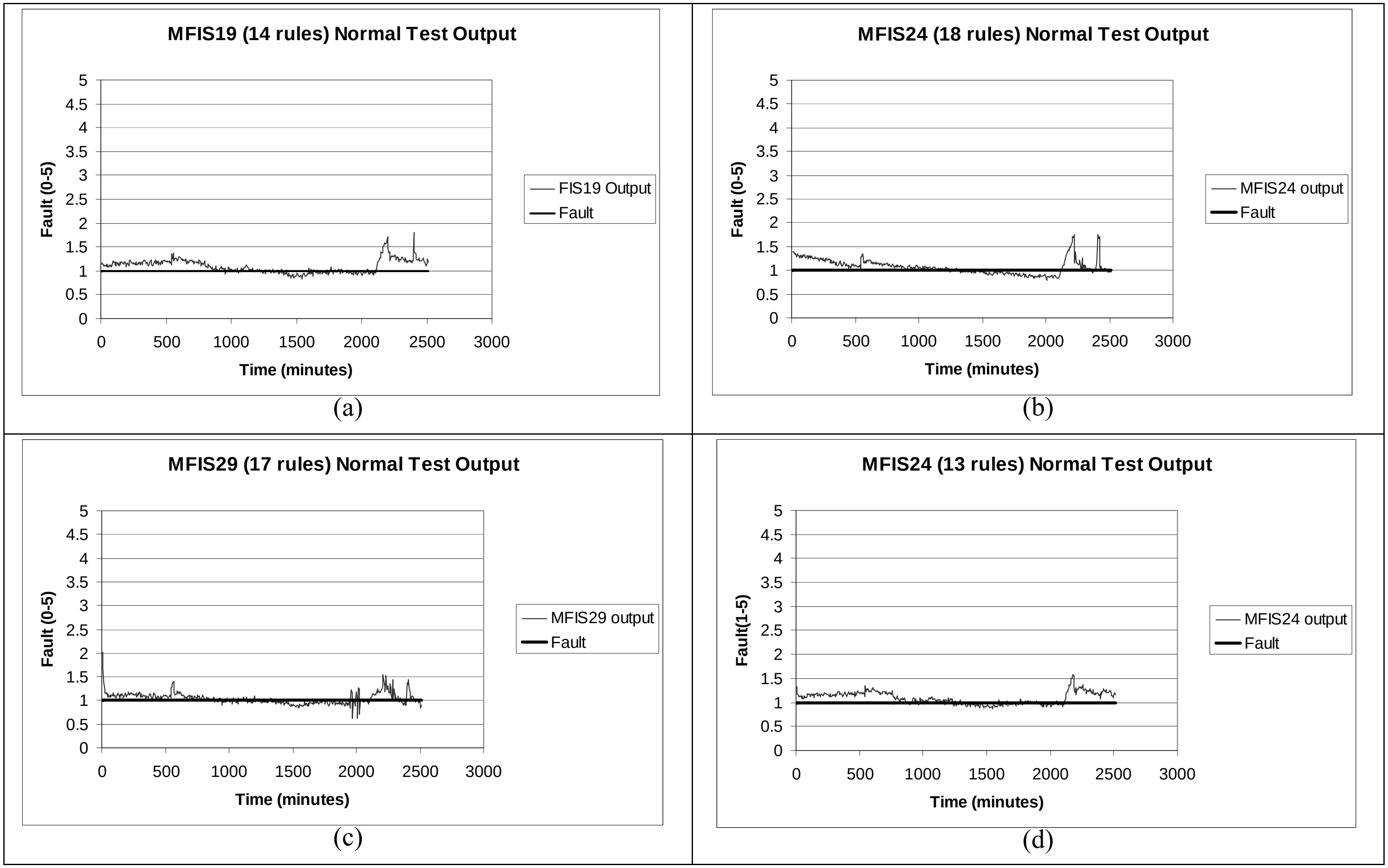

Figure 7.4 Normal Test Output for Mechanical Fault Detection System with 5-Minute Interval

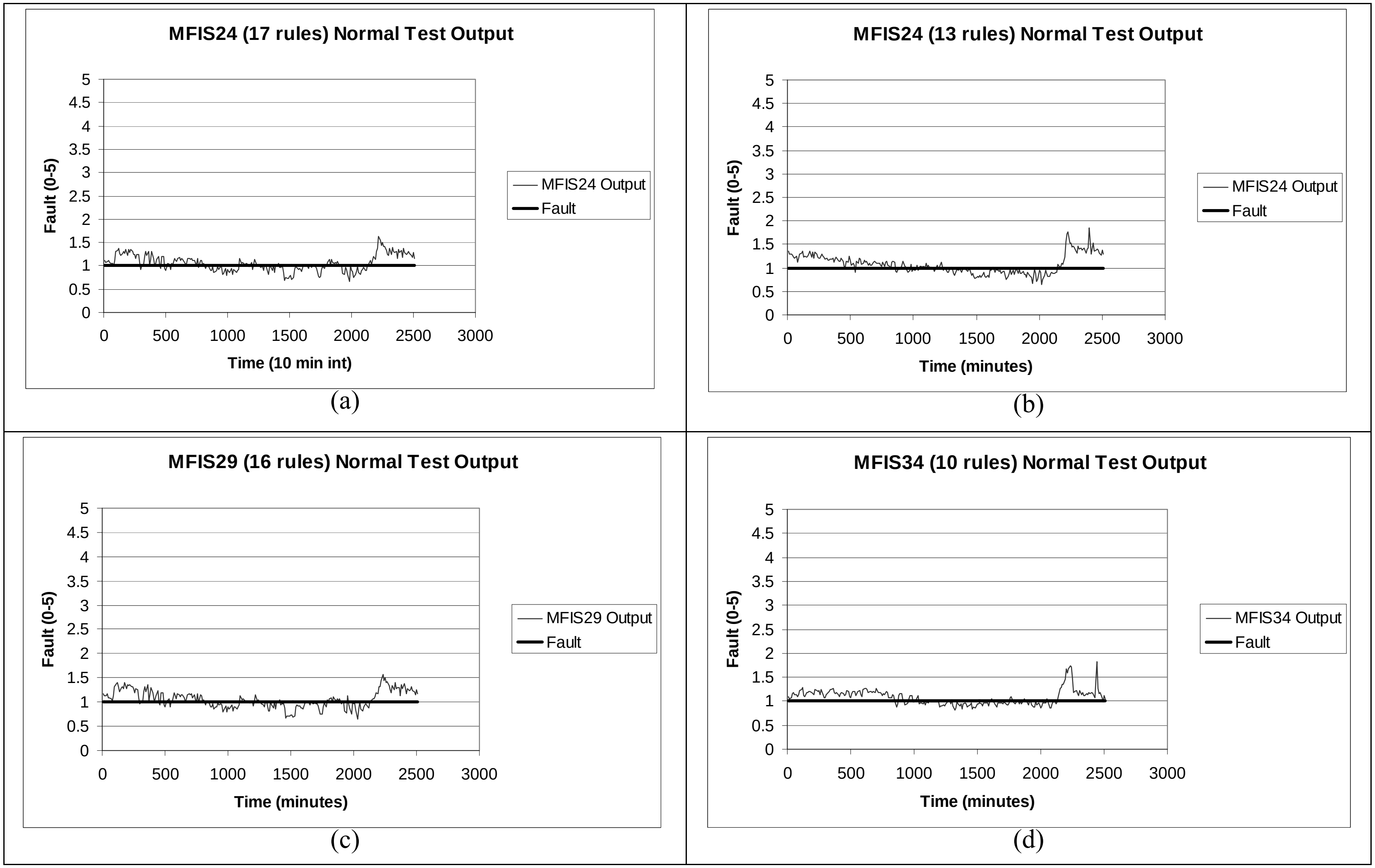

Figure 7.5 Normal Test Output for Mechanical Fault Detection System with 10-Minute Interval

The differences between the charts are slight, signaling all systems performed very well in recognizing the normal condition. The response stayed very close to 1 during most of the testing period. There was a maintenance routine at the point 109 of the FDI5 system charts that suddenly dropped the EC value slightly (about 40 μS/cm). This was a result of water addition and caused a small spike in the systems' responses.

Near the end of the responses, at the point 400-500 for the FDI5 systems and at the point 200-250 for the FDI10 systems, there were some points of the response that went farther away from the desired value. Logged data showed that during that time period there were other experiments on the other tanks that day that needed additional computer subroutine. Stopping the control and monitor program several times to include the needed subroutines caused missing sampling steps and the fault detection system recognized these as abnormal conditions.

The three experimental tanks had identical numbers of lettuce plants and nearly identical nutrient solution conditions during the experiments. Unfortunately, the evapotranspiration of the three tanks was found to be different and, as the result, the rate of the nutrient changes in the solution for each tank was different. This condition presented additional noise and reduced the overall sensitivity for the FDI system. The problem was pinpointed as the different flow rate and pattern of the airflow above the tanks. The detail information about the evapoptranspiration can be found in Appendix B. The FDI system performances for normal condition can be seen in Table 7.10

Table 7.10 FDI System Performances for Normal Condition

FDI System	Misclassification	Correct Classification
FDI5 with 24 inputs and 18 rules	1.6%	98.4%
FDI10 with 29 inputs and 16 rules	0%	100%

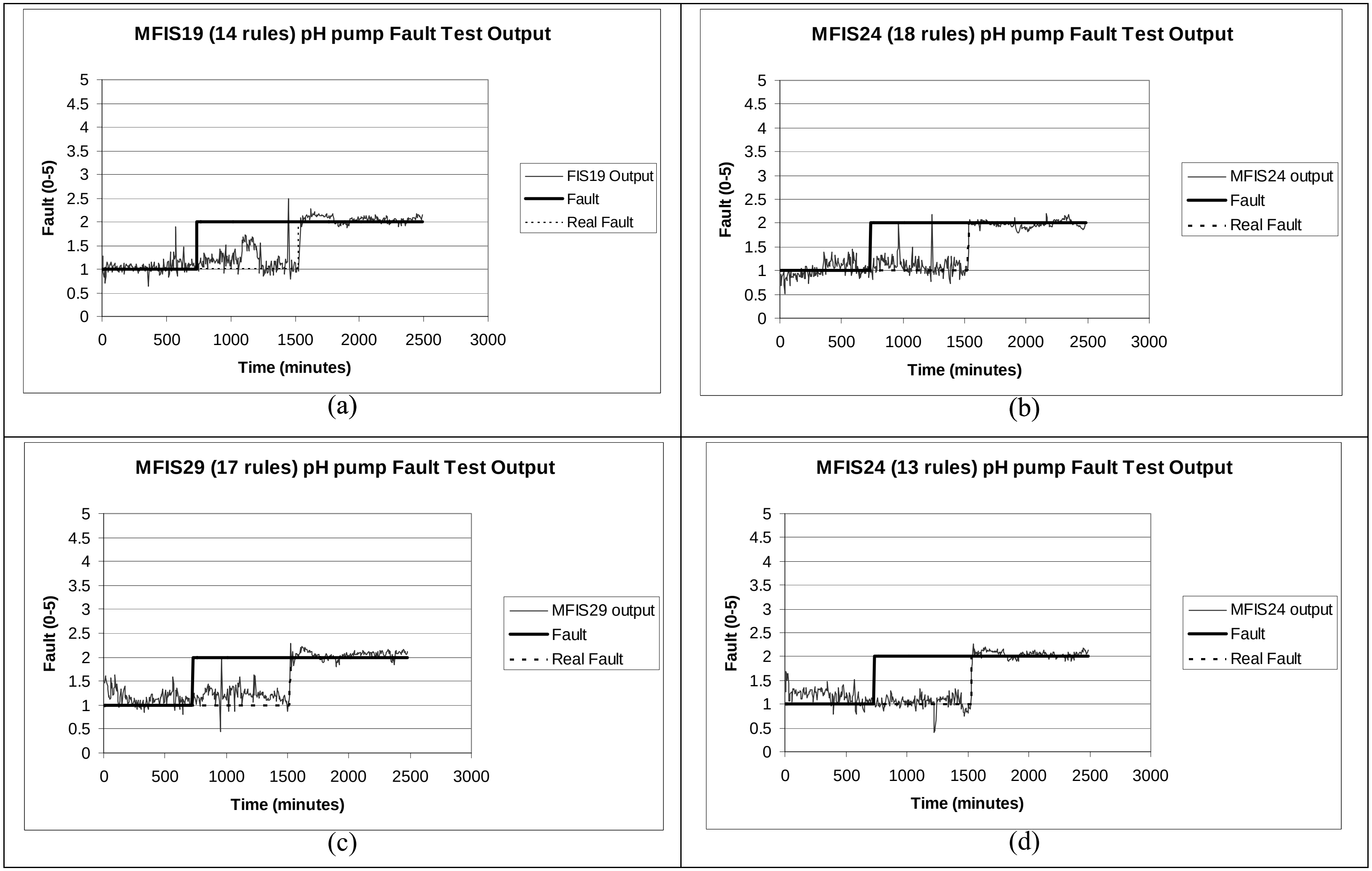

Figure 7.6 pH Pump Fault Test Output for Mechanical Fault Detection System with 5-Minute Interval

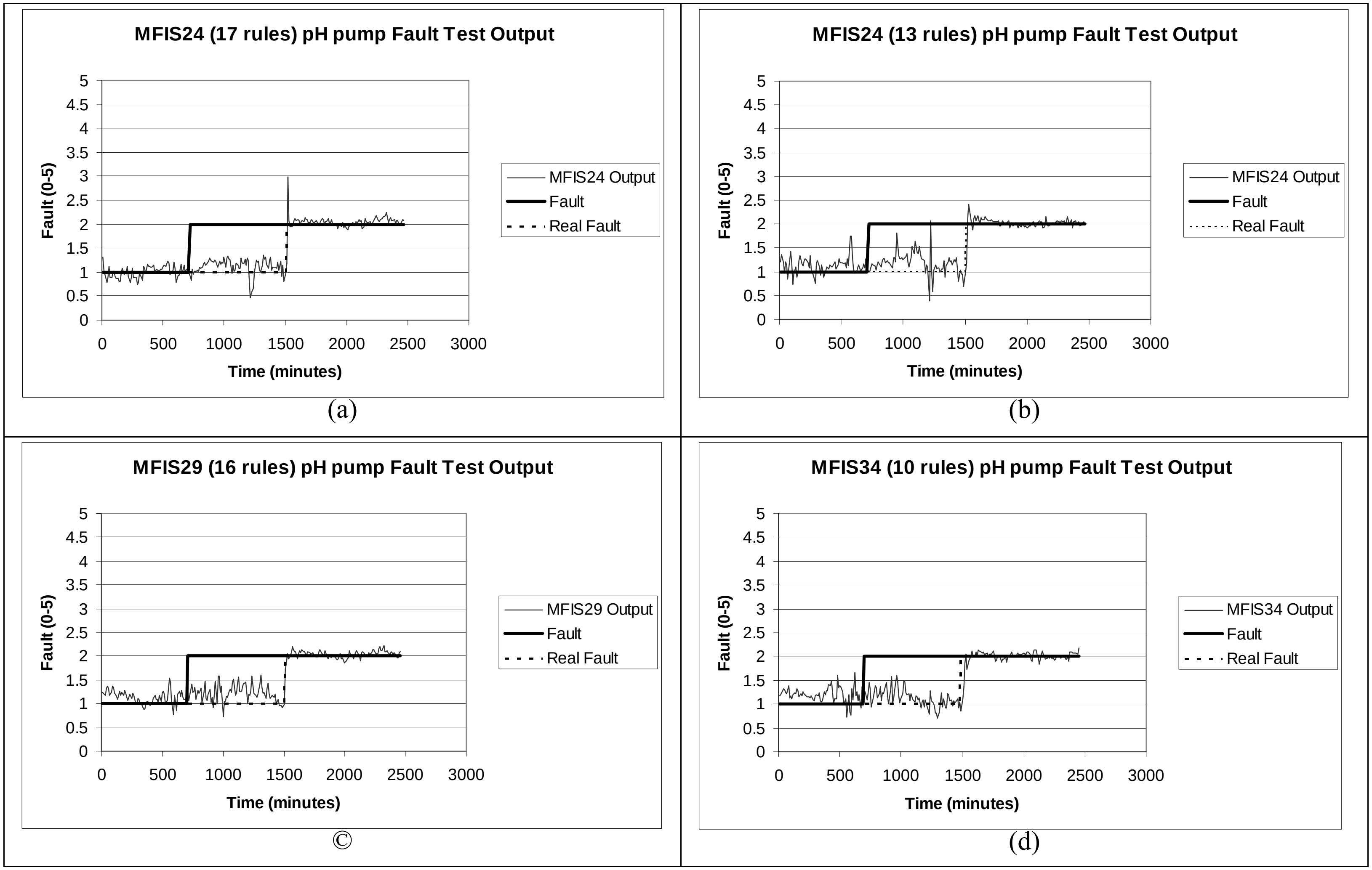

Figure 7.7 pH Pump Fault Test Output for Mechanical Fault Detection System with 10-Minute Interval

It took 795 minutes for the FDI5 systems and for the FDI10 systems to recognize the pH pump fault as shown in Figure 7.6 and Figure 7.7. This happened after the maintenance period where the pH was lower than 5.8 when the new nutrient solution was added to the tank and slowly increased until it crossed 5.8 at the minute 1530 for theFDI5 system and for the FDI10 systems.. The way the plants absorbed the nutrients (primarily nitrate) increased the pH of the nutrient solution. The pH control pump controlled the pH value by injecting an acid solution to the circulation system whenever the pH went above 5.8. This is the reason why the fault detection system did not recognize the faulty condition from the minute 735 to 1530 for the FDI5 systems and for the FDI10 systems where the pH control signal did not asked for any acid addition

The pH control pump fault training data sets were formed to recognize the fault when the pH control pump was turned off at the beginning of the fault experiment. The implications are explained below.

The FDI5 system with only 19 inputs and 8 rules can recognize this condition perfectly with 5-minute detection time. The limited degrees of freedom forced it to recognize only the faulty condition. The better FDI systems had more freedom and tried to find symptoms of the faulty condition that did not become available until the minute 1530. These FDI systems were forced to find some fault pattern during this period, which was not actually available. As the result, the FDI systems found some anomaly or noise in the data and used them as the fault symptoms. This wrong training condition showed in every pH test responses and detected as a slightly faulty condition between the minute 735 and 1530 (the FDI5 systems and the FDI10 systems) in this fault test.

The pH could stay below the set point for more than a day when a large quantity of the concentrated nutrient solution was added to compensate the effects of experiments in the different tanks. The concentrated nutrient solution needed to bring the EC value up by more than 250 μs/cm initially caused the pH to drop only from 5.8 to 5.75. The pH kept decreasing in the course of about half a day to around 5.62 before it gradually went up again to 5.8 in a 20 hour period. One of the possible explanations is that the microorganisms in the nutrient solution reacted to the sudden change in the nutrient composition and caused this change. This phenomenon is shown in Figure 7.8. At the end of the graph there is a very small dip caused by small adjustment of EC that was done regularly every two days. The pH dropped to about 5.76 and then gradually went up to 5.8 again. If the pH pump fault experiment started at the beginning of this period, the symptom would not show up before 32 hours passed.

The FDI5 systems responses have more noise than the FDI10 systems, where data averaging reduced the amount of noise in the data sets. The real fault period (starting from the minute 735 to the minute 1530) was recognized successfully by all detection systems with very little deviation. The FDI system performances for the pH pump test is shown in Table 7.11.

Table 7.11 FDI System Performance for pH Control Pump Fault

FDI System	Detection Time	Misclassification	Correct Classification
FDI5 (24 inputs and 18 rules)	5 minutes	0.1%	99.9%
FDI10 (29 inputs and 16 rules)	10 minutes	0%	100%

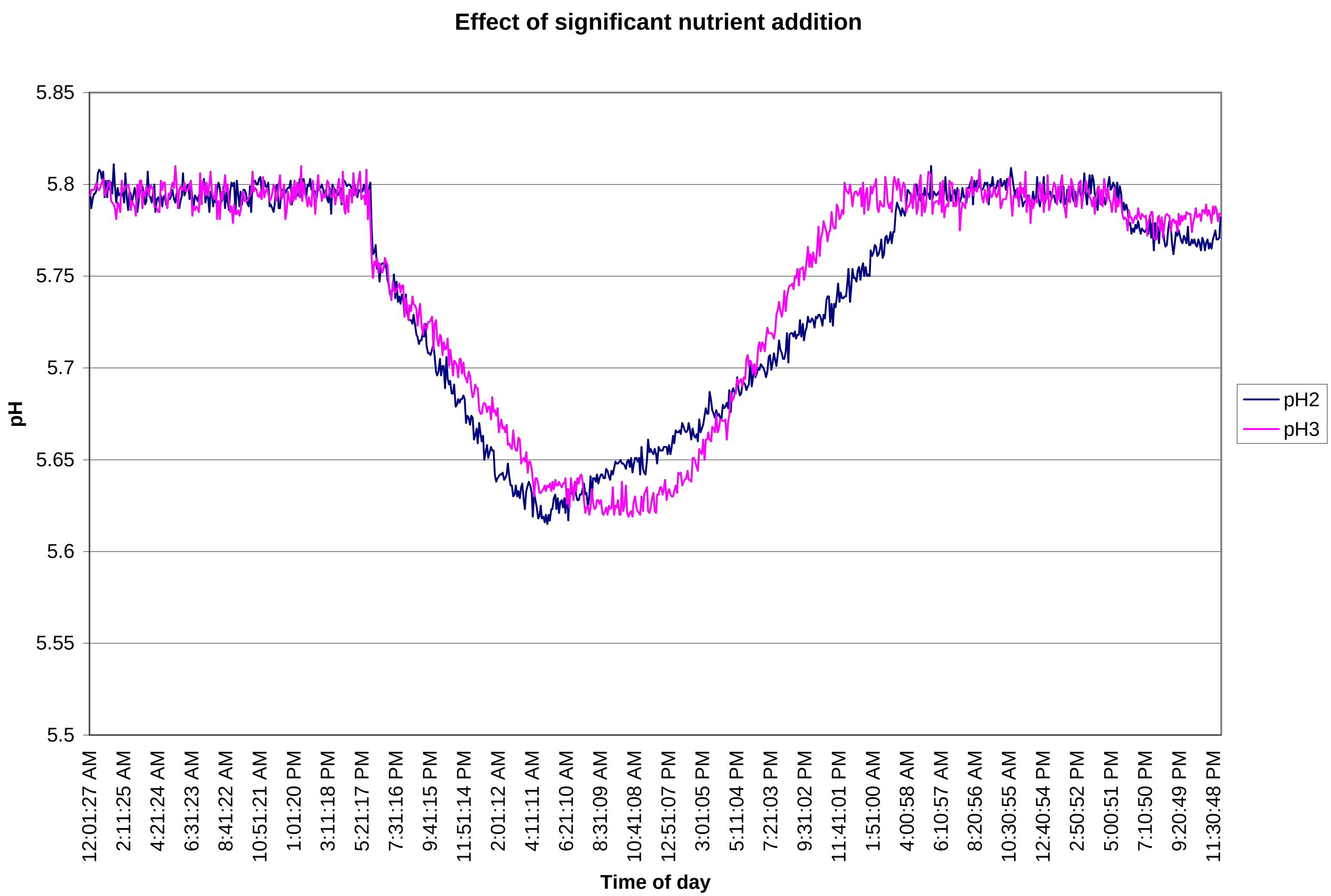

Figure 7.8 pH Drop Anomaly

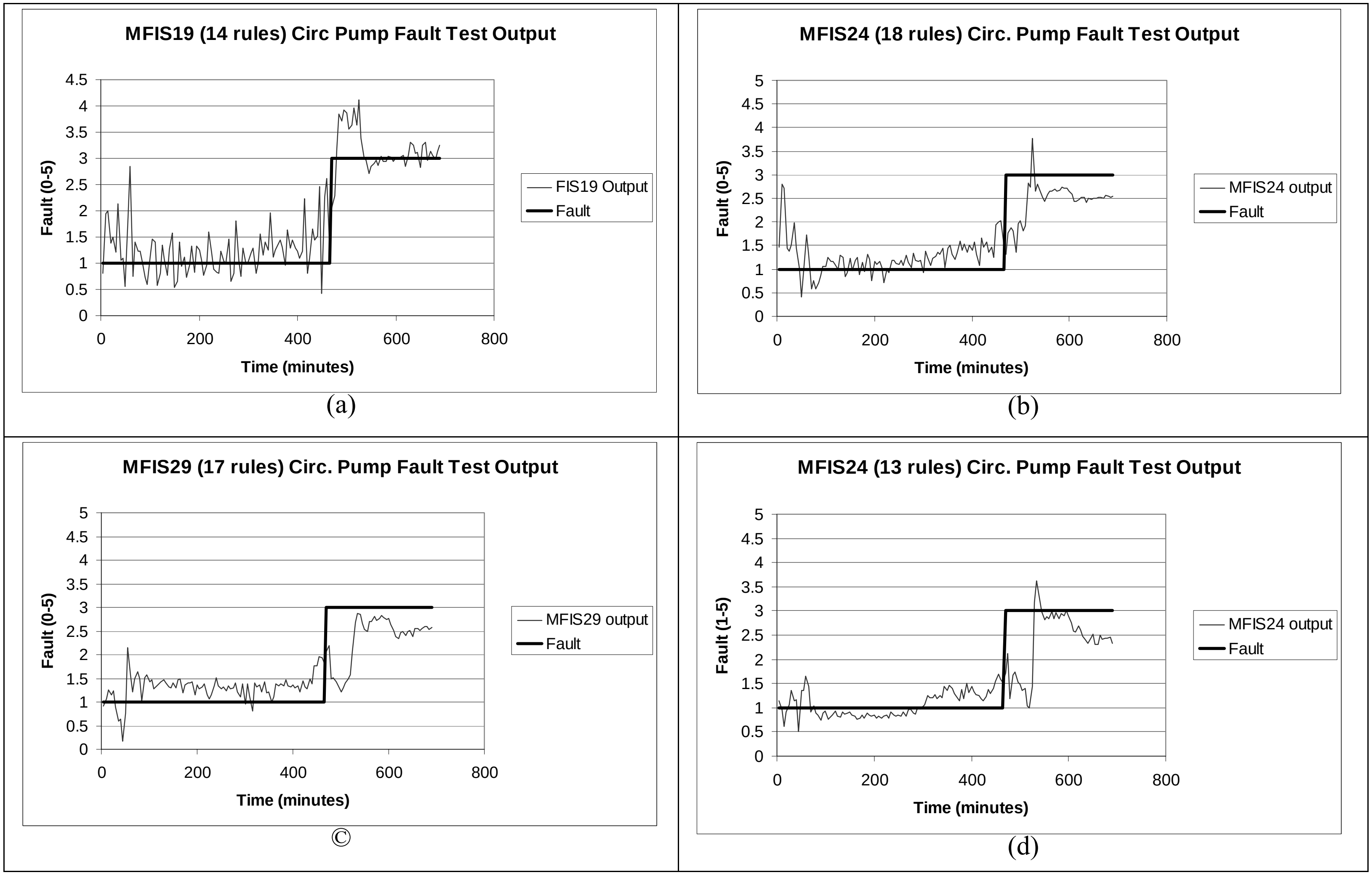

Figure 7.9 Circ Pump Test Output for Mechanical Fault Detection System with 5-Minute Interval

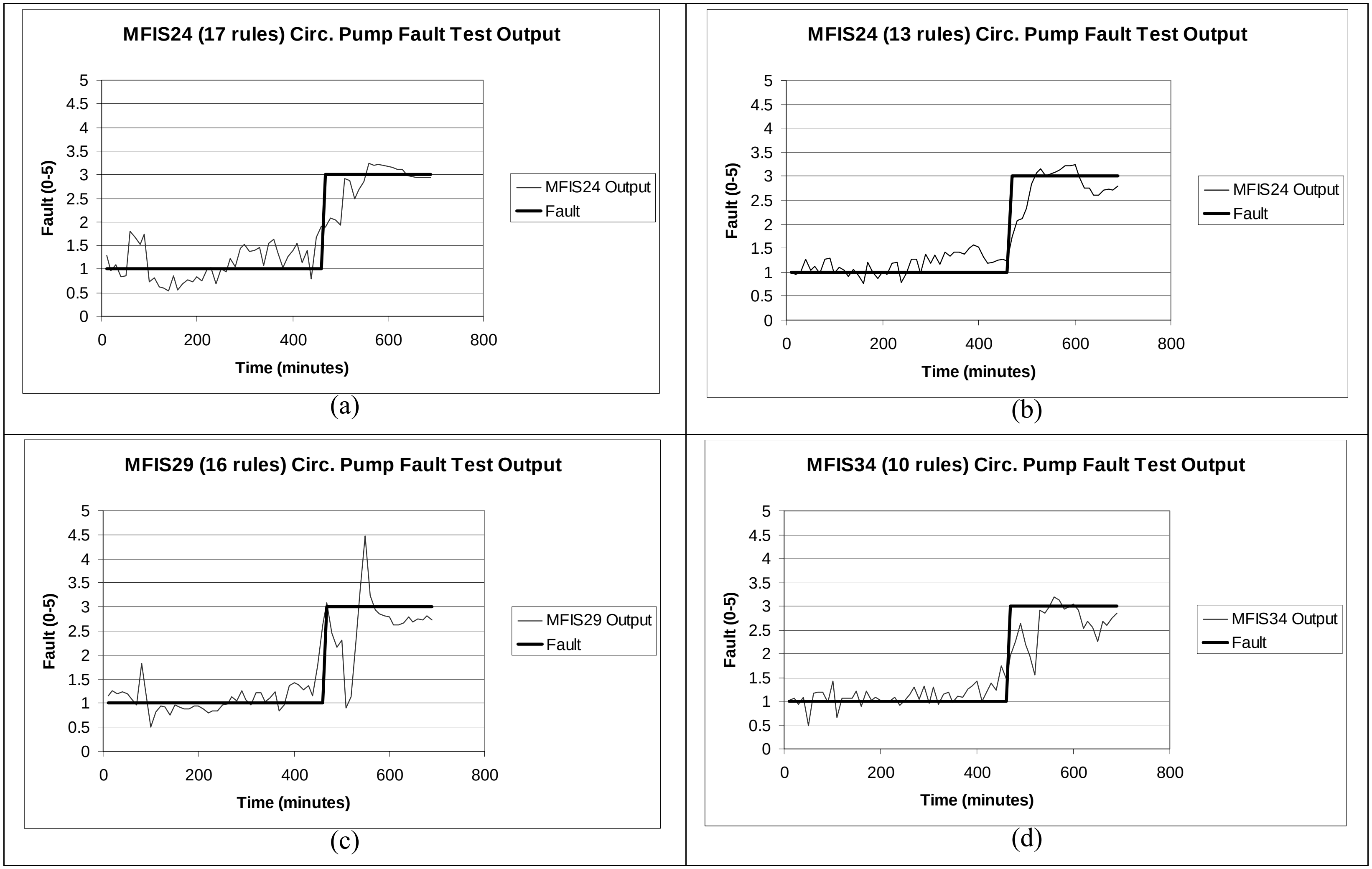

Figure 7.10 Circ Pump Fault Test Output for Mechanical Fault Detection System with 10-Minute Interval

The circulation pump fault was difficult to detect reliably using feedback from sensors especially in the beginning of the fault as shown results in Figure 7.9 and 7.10. The cessation of the nutrient solution circulation caused two problems. The first one was the noisy sensor readings. Most sensors need a minimum solution flow to replenish the solution used in the chemical reaction around the sensor membrane. The noisy reading will reduce the FDI sensitivity in detecting the fault and can cause a noisy response from the FDI. The circulation also caused a long delay for both acid injection and oxygen addition since they have to diffuse slowly through the solution instead of being quickly distributed around the tank, including the sensor location. The diffusion rates were not constant along the long path between the injection sites and sensor location so each circulation pump will have different diffusion rates.

Depending on the state of the nutrient solution, there were two sets of variables that are important. One was pH and pH control; the other was DO and DO control. If the DO value was low in the beginning of the fault and the DO control was on, the DO would keep decreasing even if the DO control increased. If the DO concentration was well above set point in the system, there was no DO control signal feedback for the circulation pump fault.

The pH value can also be used to detect this fault. If the pH value was nearly or already 5.8 and the pH control pump gave away small amount of acid to keep it at this value, the sudden stop of the circulation pump prevented the acid addition reaching to the sensor location. The pH control increased the acid injection without any effect on the pH reading at the sensor for a long time. The pH dropped significantly after the slowly diffusing acid in the nutrient solution reached the pH sensor location. The pH control would suddenly stop the acid injection although it was too late. The pH would have dropped to around 3-4 which was low enough to do some damage to the roots. To reduce roots damage problem, the circulation pump fault experiments were kept to be as short as possible. In the beginning of the fault, the symptom detected from the pH value was the same as the symptom detected from the pH control pump. The difference is after a long period of time the pH suddenly decreased by the time the diffused acid injection reached the sensor location.

For this particular test, the DO control opened the solenoid valve to let some pure oxygen dissolve into the circulation pipe at the minute 445. It took between 5-10 minutes for the oxygen to reach the sensor location, and more time was needed for the oxygen to reach the top limit of 7 mg/l. The DO control was on for 20 minutes (four data sampling points). During this delay, the DO value kept dropping and the FDI5 system output rose to about 2, two thirds of the way to the circulation pump fault value of 3. At this time the DO value had already increased but since the desired max limit value was still not reached, the DO signal was remained on. The circulation pump was intentionally turned off at minute 470 signaling the beginning of the circulation pump fault experiment. At this point the DO dropped faster than its usual rate and the DO sensor reading was erratic. The pH sensor had the same behavior. This condition changed at minute 525 where the DO value decreased so much, the noise was insignificant and the DO controller started asking for the DO addition continuously. At this point the FDI systems finally detected the condition as a circulation fault after a few sampling steps following minute 525.

The FDI can definitely detect a circulation fault if the symptoms from the DO or both the DO and the pH are positive. If the only symptom available was from pH, the symptom was similar to a pH control pump fault and the FDI system needed more symptoms to detect it correctly.

The normal portion of the test responses was very noisy but for most of the time stayed below 1.5 and averaged around 1, which can still be defined as normal. The FDI systems responded differently to the chaotic period between the minute 470 and the minute 525. Figure 7.9a (FDI5) shows the response jumped to a value of 4 during this period while the other responses stayed between 1 and 2. The FDI system response in Figure 7.9b detected the fault successfully as a circulation fault after the minute 525 and the response value stayed around 3. The other charts had response values between 2.5 and 3, signaling that some of the rules in the FDI systems considered this as the pH control pump fault symptom and the aggregate output was down a bit although still considered as circulation pump fault. The FDI10 systems responses are similar to the FDI5 system with a little less noise caused by data averaging.

The FDI system for this fault can be formed better with adding a flow sensor somewhere along the pipe. As soon as the flow stops the flow rate will be around zero and the FDI can easily recognize this as the circulation pump fault. Although additional costs are involved, it can be justified based on the irreversible effect of the destruction to the plants roots caused by the excessive acid addition. The FDI system performances can be seen in Table 7.12.

Table 7.12 FDI System Performances for Circulation Pump Fault

FDI System	Detection Time	Misclassification in Fault 3	Correct Classification
FDI5	45 minutes	2.2%	97.8%
FDI10	60 minues	4.3%	95.7%

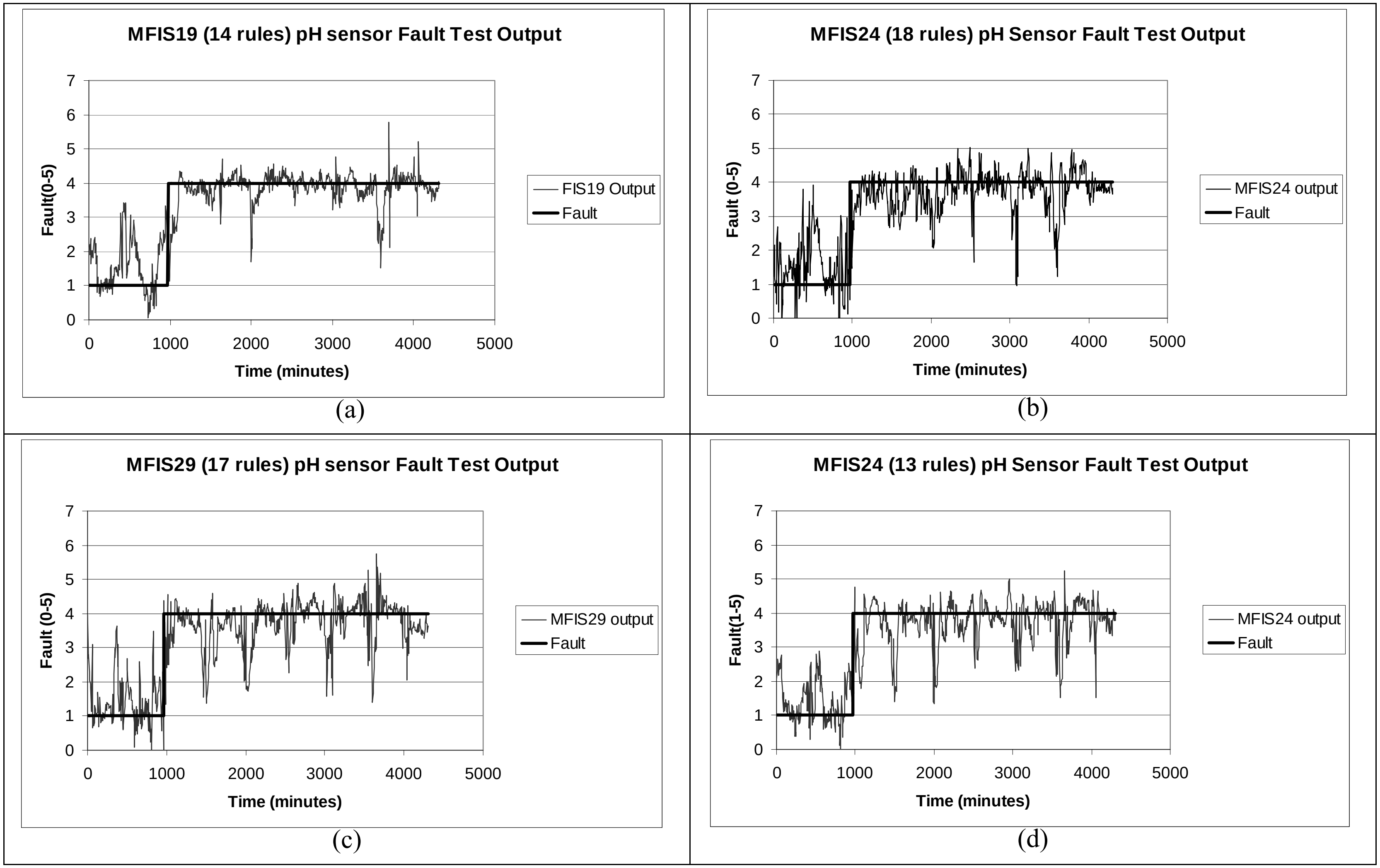

Figure 7.11 pH Sensor Fault Test Output for Mechanical Fault Detection System with 5-Minute Interval

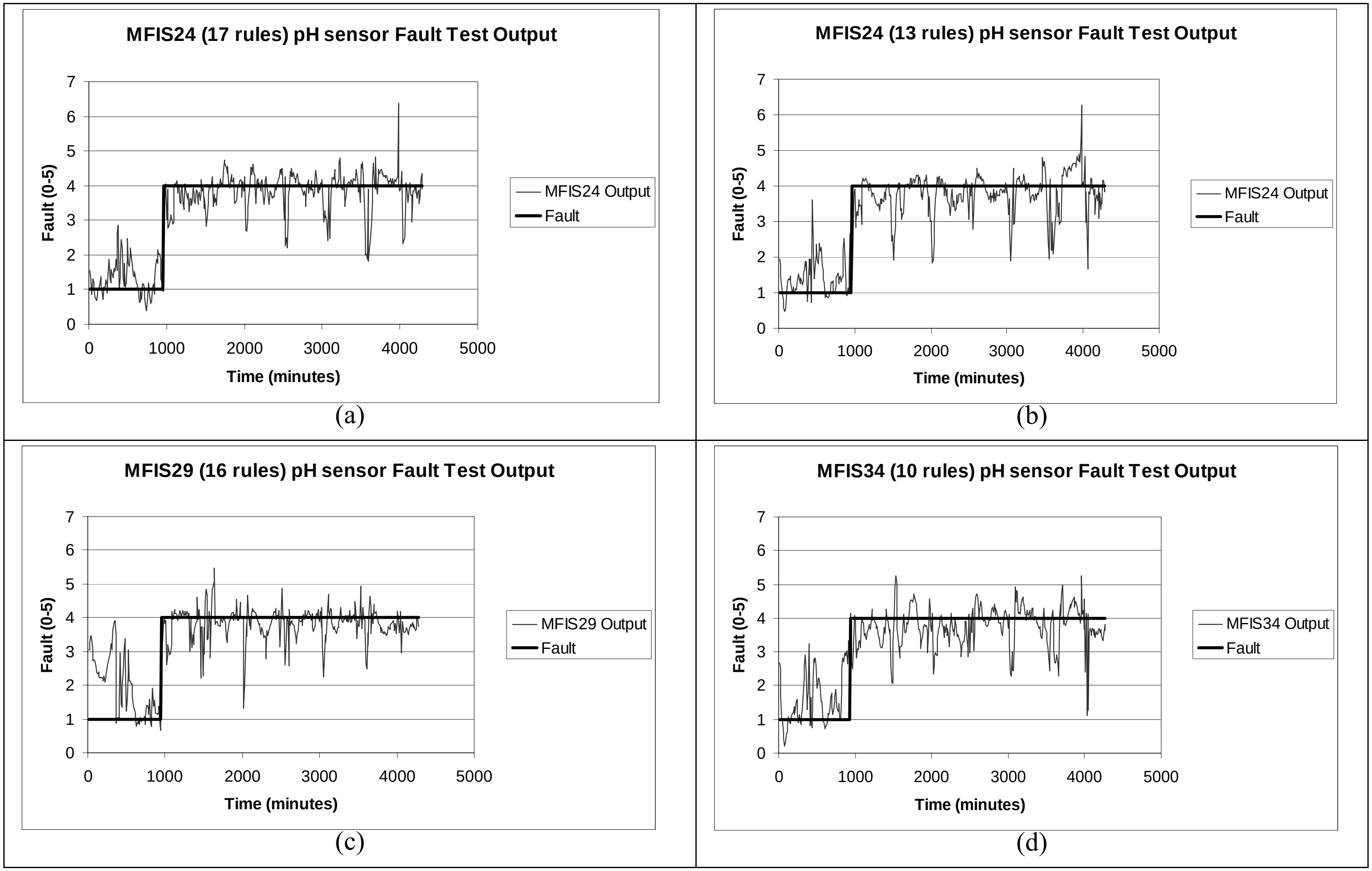

Figure 7.12 pH Sensor Fault Test Output for Mechanical Fault Detection System with 10-Minute Interval

Sensors in general have a tendency to drift away from their calibrated state. The drifting can be linear or non-linear depending on the type of sensor. The non-linear drifting is harder to detect correctly. The pH sensor fault test used a slowly changing sine noise with amplitude of 1 and period of 525 minute to simulate the drifting. The sine wave changed from the maximum value of 5.8 to the minimum value of 3.8, simulating the pH sensor reading drifting down and drifting up. The pH value drifting with sine wave noise is shown in Figure 7.13. The peaks of the sine wave were clipped in the chart. The real pH value increased slowly during the test time so by the time the simulated pH reached the top of the wave, it was above the set point. The pH control injected some acid to correct this condition. This nonlinear behavior added some noise for the detection system to overcome. The pH control was zero for all other parts of the test.

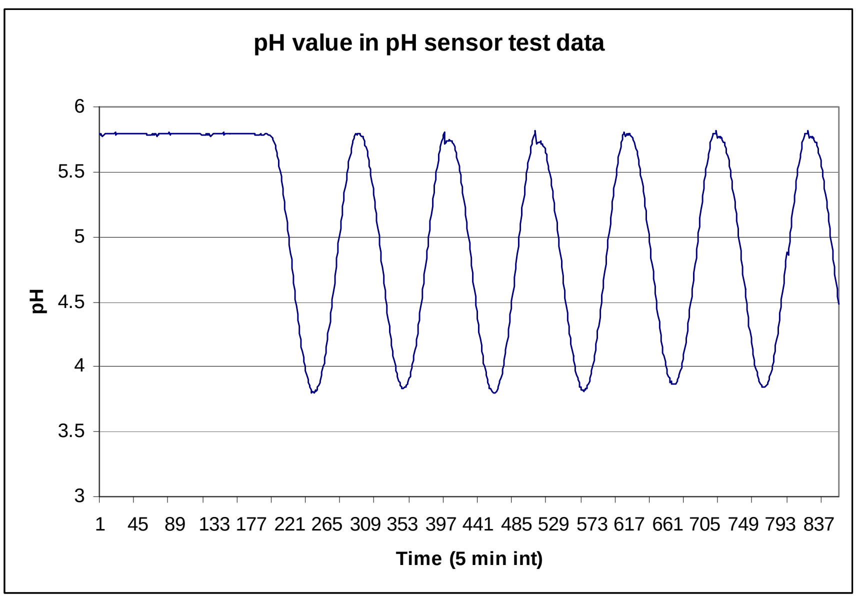

Figure 7.13 pH Values with Sine Wave Noise in pH Sensor Test Data

Figure 7.11 and Figure 7.12 shows the pH sensor fault test result for the FDI5 and FDI10 systems. The normal parts of the responses did not always follow the intended values. This problem surfaced since the conditions of the three tanks were not completely identical with each other, as mentioned in the normal condition test discussion. The FDI10 systems had less noisy responses and had values closer to normal.

The FDI systems identified the drifting immediately and the fault value went up to 4. The symptoms for this fault was gradually weakened since the pH value drifted up again to the normal condition in a sine wave period of 105 sampling steps after the error was introduced. The fault value went down to near 1 before going up to 4 again following the sine wave noise. This behavior periodically happened in the FDI result as the value of the pH drifting up and down again. The FDI system performances can be seen in Table 7.13.

Table 7.13 FDI System Performances for pH Sensor Fault

FDI system	Detection time	Missclassification				Correct classification
		Normal	Fault 1	Fault 2	Fault 4	
FDI5	90 minutes	0.8%	1.9%	16.3%	4%	77%
FDI10	30 minutes	0.3%	0.9%	12%	0.6%	86.2%

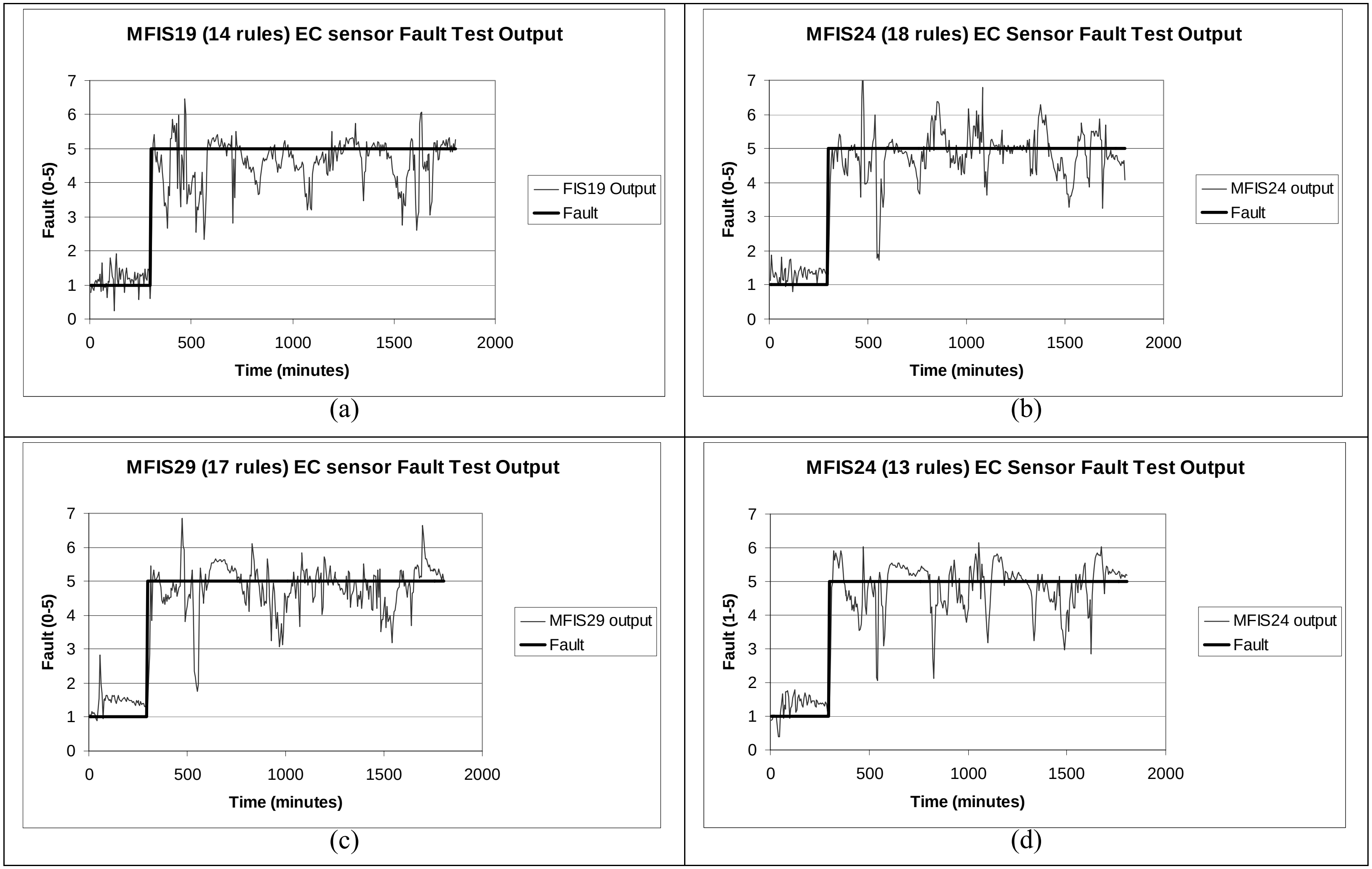

Figure 7.14 EC Sensor Fault Test Output for Mechanical Fault Detection System with 5-Minute Interval

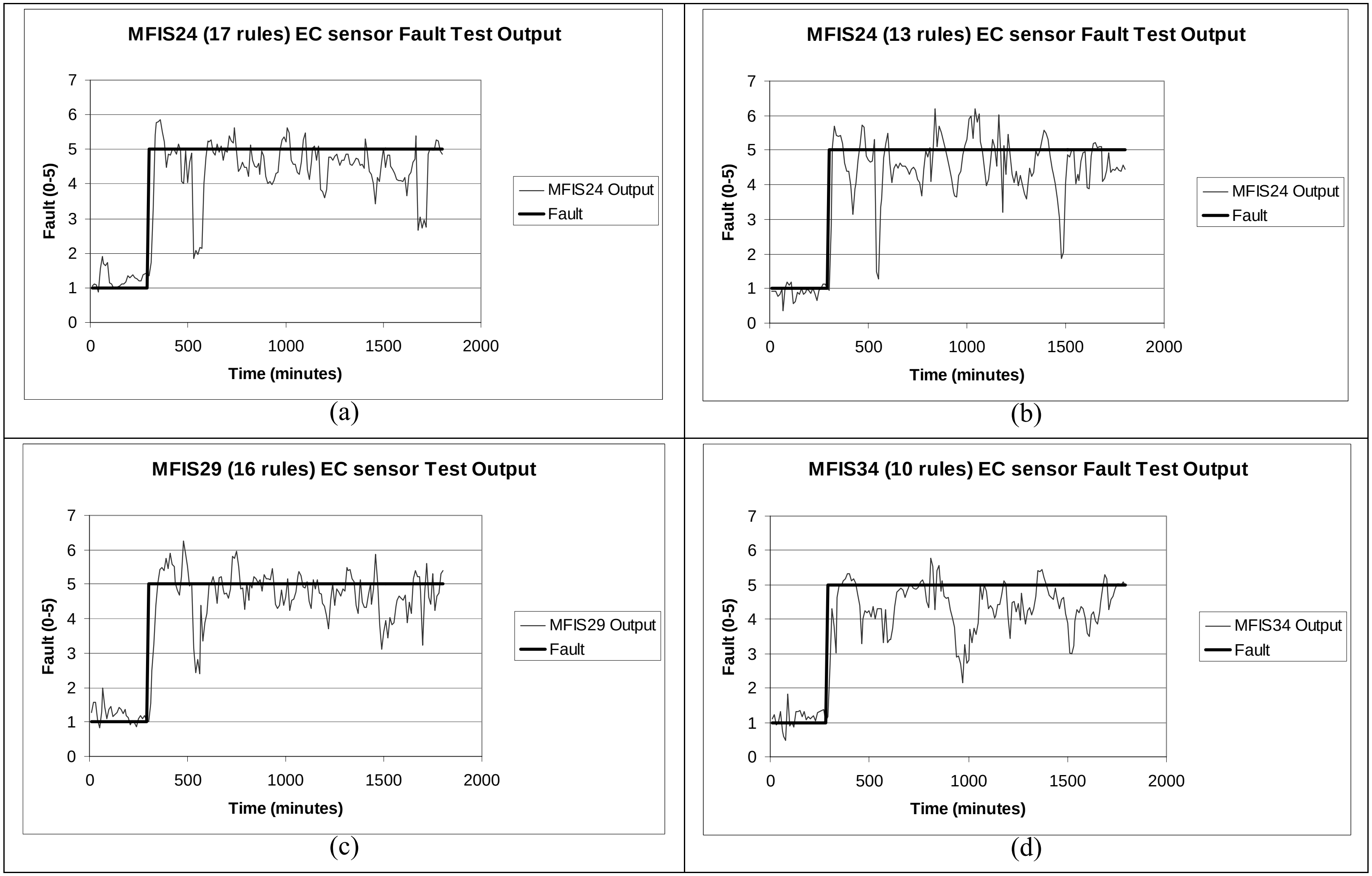

Figure 7.15 EC Sensor Fault Test Output for Mechanical Fault Detection System with 10-Minute Interval

The simulated EC sensor fault was similar to the pH sensor fault. The difference can be found by comparing Figure 7.13 and Figure 7.16. The value of the EC slowly increased from 1243 μS/cm to 1543 μS/cm and slowly decreased back to the true value of 1243 μS/cm. It kept decreasing to reach the lowest value of 901 and then increased again, following the sine wave.

The sine wave noise in EC values was smooth except for the first wave. The sine noise started at the minute 305 where the EC value was 1243 μS/cm. Since the amplitude was 300 μS/cm, the top value was 1543 μS/cm as shown at the first wave top at the minute 440. Suddenly at the minute 470 the EC value that was 1519 at the minute 465 dropped to 1464. The drop was caused by a maintenance routine that added water and concentrated nutrient solution for maintaining water level and EC. In this particular maintenance, only water was added without EC adjustment. The sudden drop in EC caused a spike in the EC sensor FDI systems response at the minute 470 as shown in Figure 7.14.

The result of EC sensor fault test was better for the normal part of the test compared to the pH sensor test as shown in Figure 7.14 (FDI5) and Figure 7.15 (FDI10). The faulty condition part of the test responses was also less noisy than the pH response. Since the EC value goes up and down as a sine wave with the set point value as the zero value for the sine wave, the set point value was reached every 270 minutes instead of 525 minutes as in the pH sensor fault.

The FDI systems recognized the normal condition satisfactorily for this EC sensor fault. It also detected the faulty condition better since it recognized the normal condition better with less noise. The fault value dropped to 4 when the sine wave noise was around zero and the EC value equaled the set point.

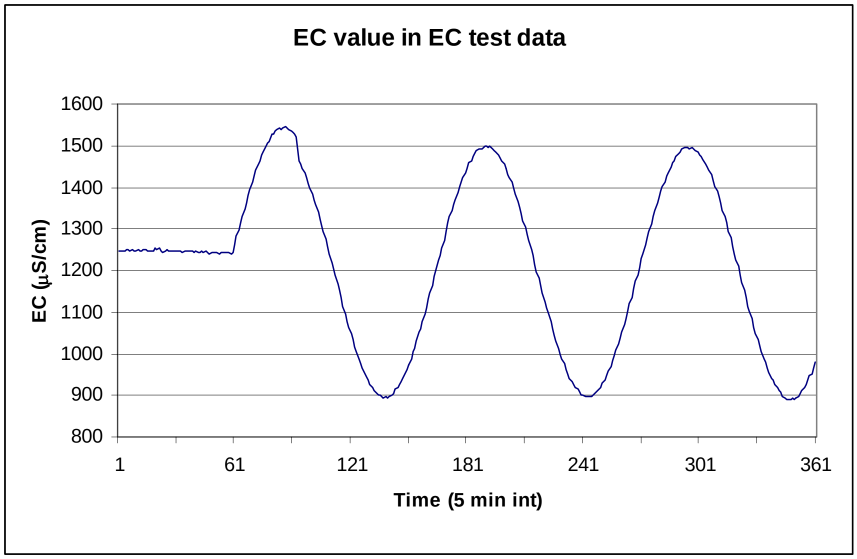

Figure 7.16 EC Value with Sine Wave Noise in EC Sensor Test Data

The FDI10 responses were less noisy than the ones for FDI5 systems. The 10-minute data had less noise compared with the 5-minute data set. The faulty condition was identified most of the time with some misclassification for fault values of 3 and 2 and many for a fault value of 4. This noisy condition was caused by the sine wave simulation of the EC sensor drifting where the value drifted up and down around the real EC value. The FDI system performances for the EC sensor fault are shown in Table 7.14.

Table 7.14 FDI System Performances for EC Sensor Fault

FDI system	Detection time	Misclassification				Correct classification
		Normal	Fault 1	Fault 2	Fault 3	
FDI5	15 minutes	0%	1%	1.3%	14.9%	82.8%
FDI10	50 minutes	0%	0%	5.3%	15.3%	79.8%

7.8 Result Summary

Table 7.15 Detection Times of NF Mechanical FDI Systems

	10min	**20min**	**30min**	**40min**	**1h**	**1.5h**	**6h**	**8h**	**13h**
Fault 1	0%	0%	100%	100%	100%	100%	100%	100%	100%
Fault 2	0%	100%	100%	100%	100%	100%	100%	100%	100%
Fault 3	0%	0%	100%	100%	100%	100%	100%	100%	100%
Fault 4	0%	100%	100%	100%	100%	100%	100%	100%	100%

Detection time for the NF FDI systems for fault 1 to fault 4 are shown in Table 7.15. All faults were detected within 30 minutes from the beginning of the faults. This fast detection gives enough time for grower to correct the situation before it affects the plant quality.

Table 7.16 Classification Percentages of NF Mechanical FDI System Responses

Tested data set	**Classification in:**				
	Normal	**Fault 1**	**Fault 2**	**Fault 3**	**Fault 4**
Normal	99%	1%	0%	0%	0%
Fault 1	0%	100%	0%	0%	0%
Fault 2	0%	0%	96.7%	3.3%	0%
Fault 3	0.6%	1.4%	14.2%	81.6%	2.3%
Fault 4	0%	0.5%	3.3%	15.1%	81.3%

The normal condition can successfully be recognized by the NF Mechanical FDI systems with 99% correct classification as can be seen in Figure 7.14. The Fault 1 (the pH control pump fault) has 100% classification. The FDI system can identify the fault pattern accurately. The fault 3 and the fault 4 are periodic faults. They drifted to one direction slowly and drifted back slowly to another direction. These faults represented incipient and intermittent faults in the system. The intermittent nature of being normal in an instant and faulty at another instant gave a big challenge to the FDI systems. They correctly detected the faulty condition but they failed to correctly classify the faults for 22.5 % and 49.4% of the time. By repositioning both periodic faults to levels next to the normal condition, the correct classification percentage can jump to 99.4% for the “Fault 3” and 98.6% for the “Fault 4”.

CHAPTER 8

NEURO-FUZZY AND NEURAL NETWORK FDI SYSTEM COMPARISON

A Multi Layer Perceptron (MLP) Neural Network with one hidden layer was used to detect faults in the same system used in this research (Ferentinos 2002). The neural network (NN) FDI system response charts start from the beginning of the fault experiment so the comparisons are made from this point. The same time periods are observed for each methods. NN system performance definition is the same as the one for the neuro-fuzzy biological FDI system described in section 6.5.

8.1 Biological Fault Responses Comparison

Three test files were used to compare the neural network (NN) and the neuro-fuzzy (NF) FDI systems. The first two files represented transpiration fault experiments and the last one represented a normal condition.

Test 1 (transpiration fault responses from the NN and the NF FDI systems can be seen at Figure 8.1.and Figure 8.2. Both responses are noisy with the NN response oscillating more between the faulty (value of 1) and the normal conditions (value of 0). The NF also oscillates with the same pattern but with less amplitude. Periodic maintenance activities had a large impact on the NN response and a smaller impact to the NF FDI system. The detection time for the NN was about 180 minutes and for the NF is only 50 minutes. The main cause for the long detection time for the NN was the inability for the NN to reduce the effect of noise in the system. The correct classification is about 65% for the NN and 99% for the NF.

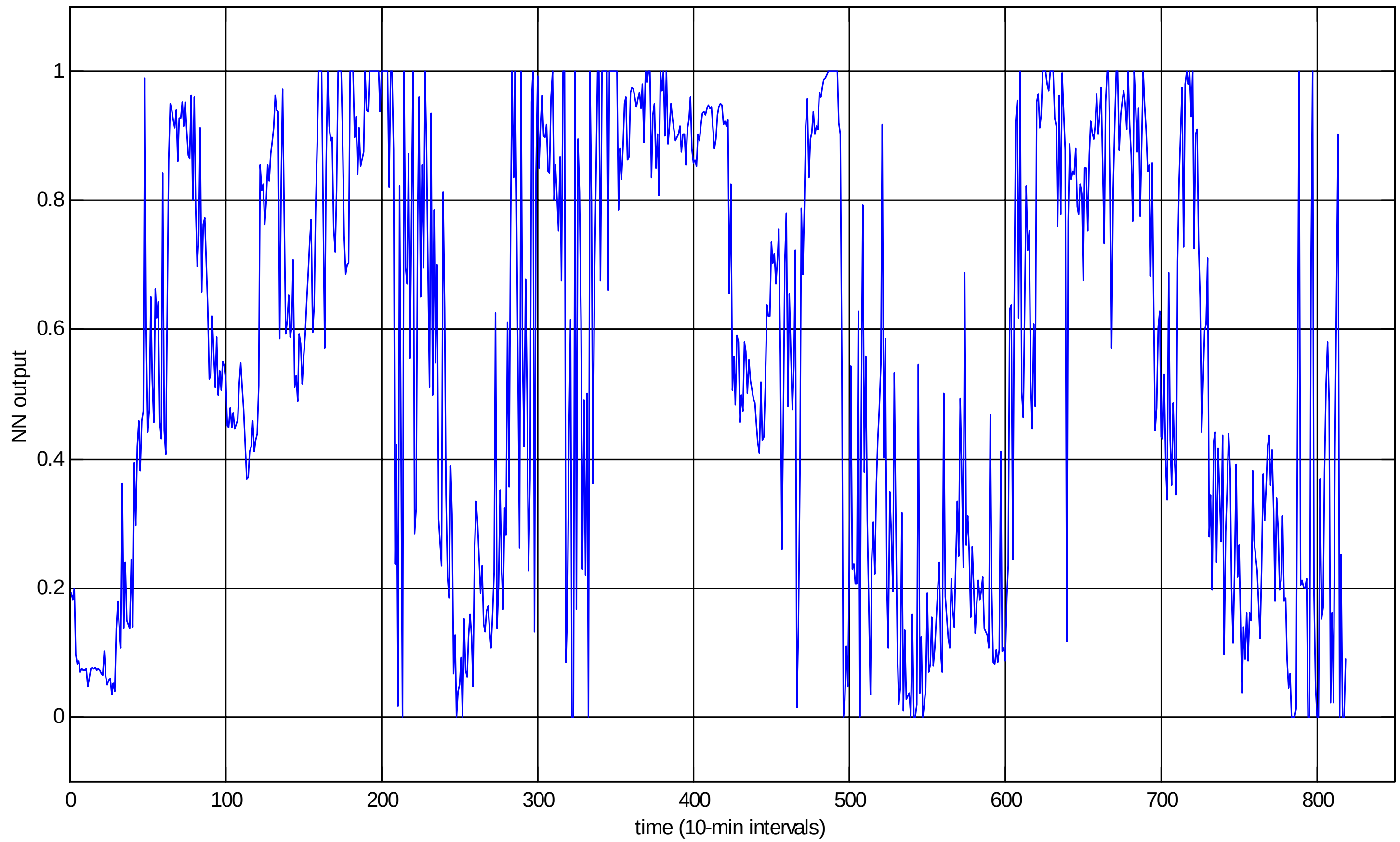

Figure 8.1 The NN (courtesy of Ferentinos, 2002)) Output During The First Testing Data Set (Faulty Operation)

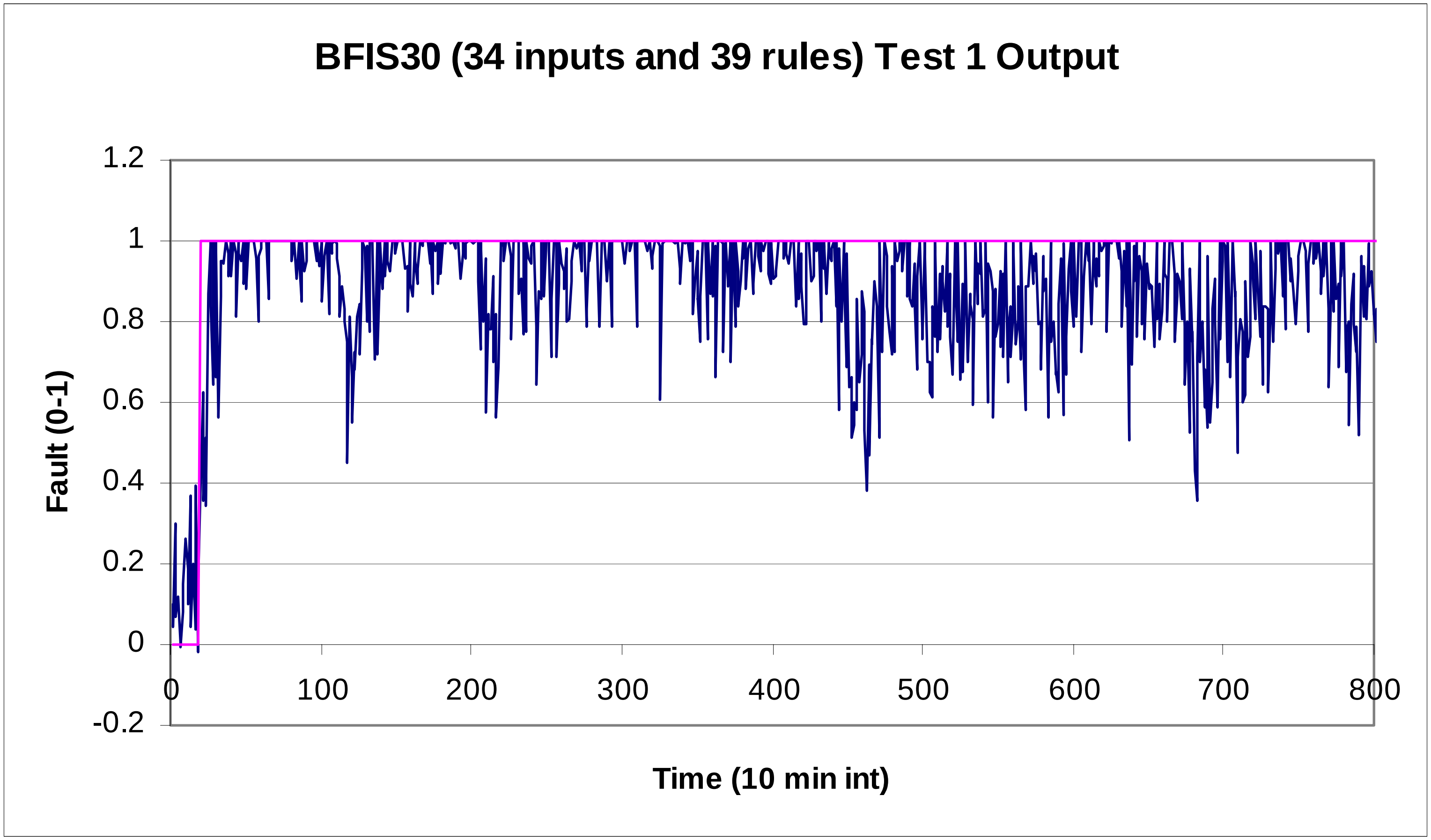

Figure 8.2 The NF Output During The First Testing Data Set (Faulty Operation)

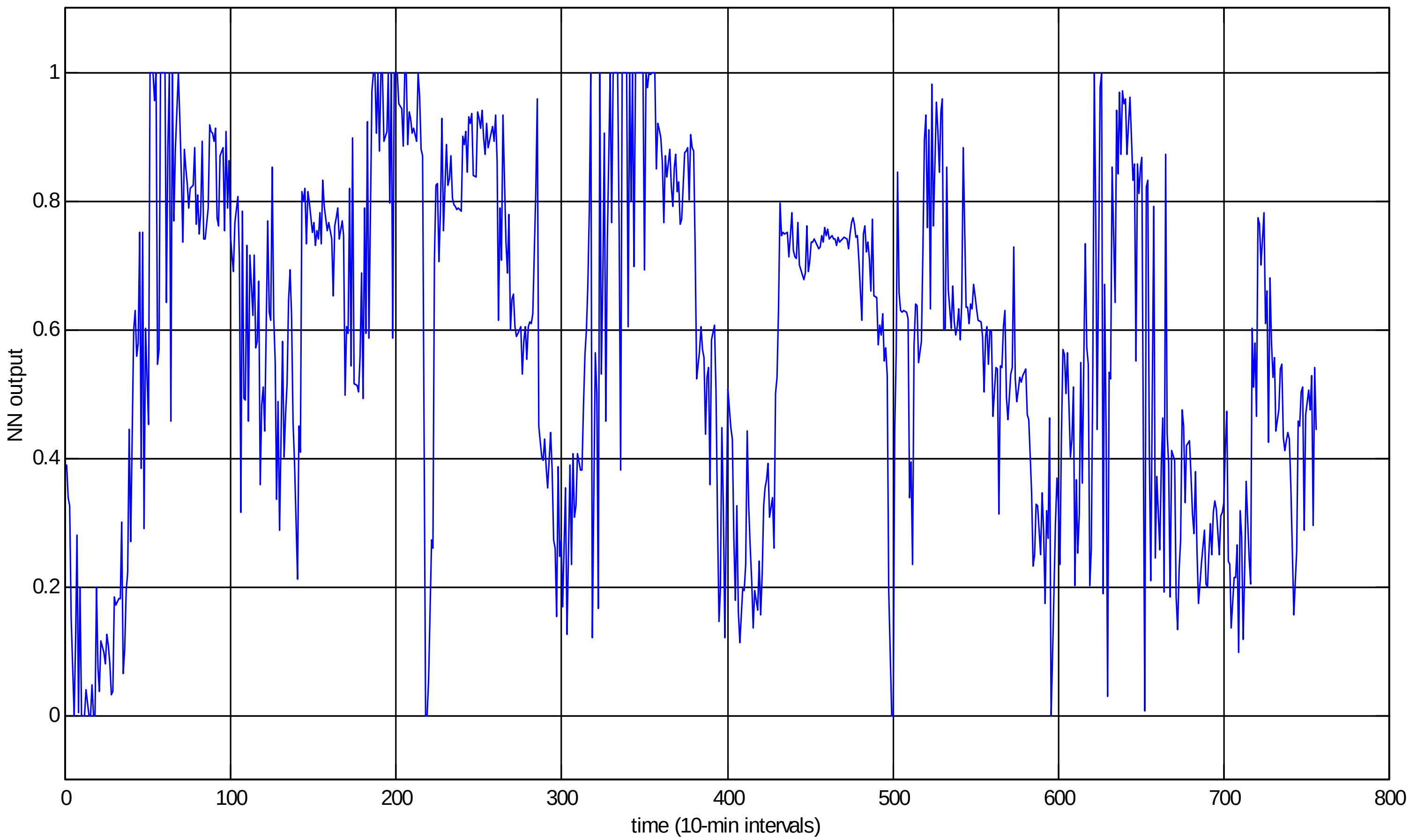

Figure 8.3 The NN (Courtesy of Ferentinos, 2002) Output During The Second Testing Data Set (Faulty Operation)

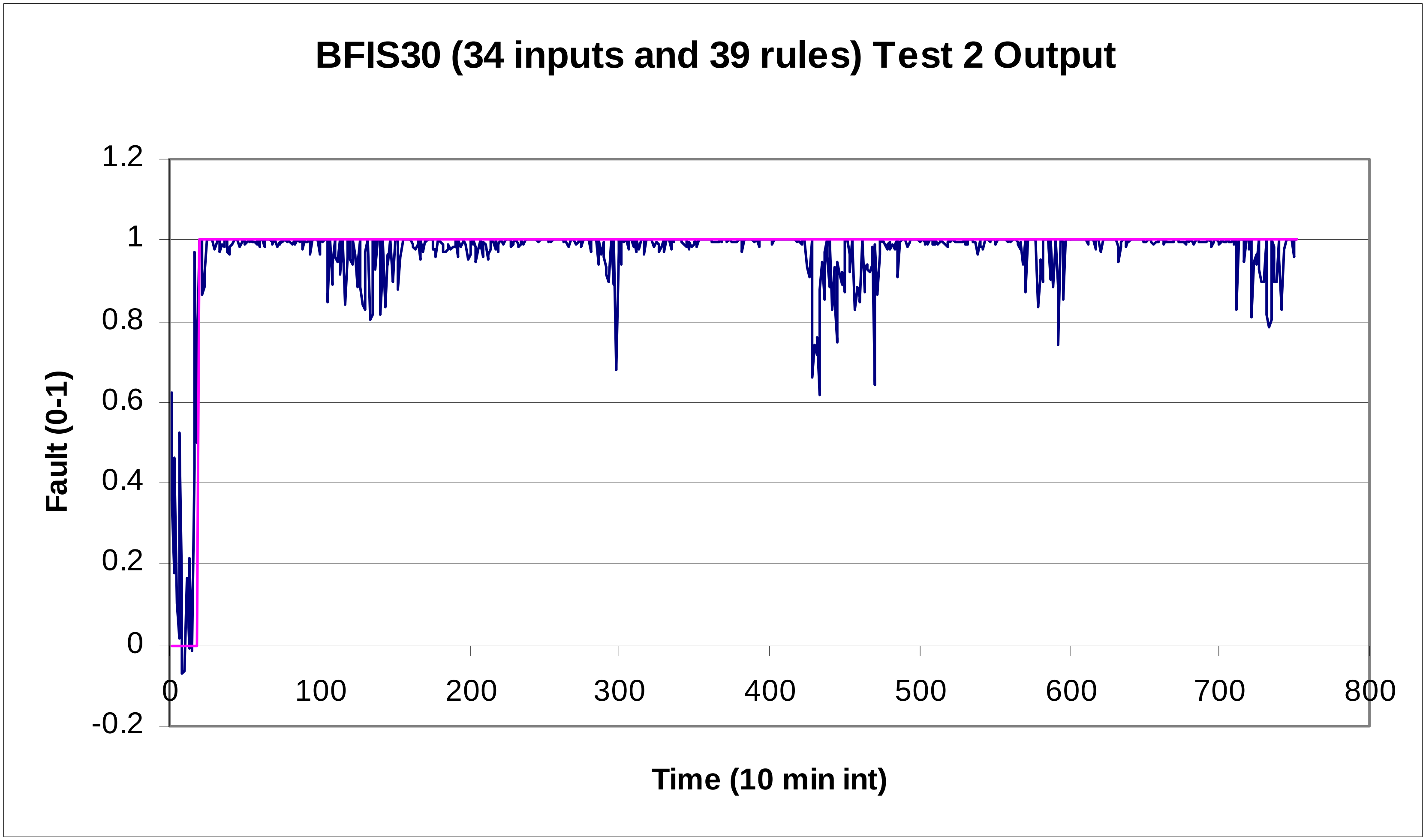

Figure 8.4 The NF Output During The Second Testing Data Set (Faulty Operation)

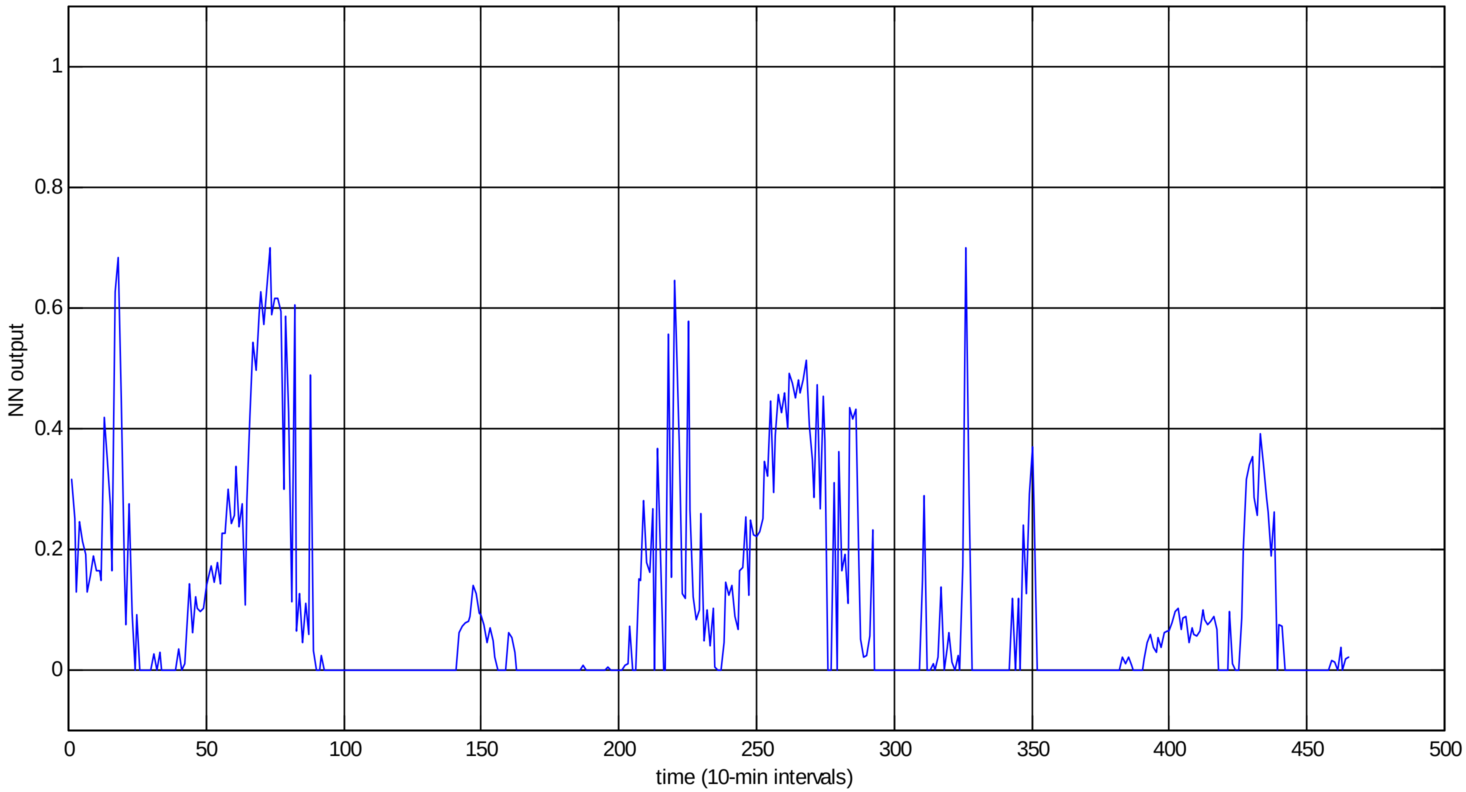

Figure 8.5 The NN (courtesy of Ferentinos, 2002) Output During The Third Testing Data Set (Normal Operation)

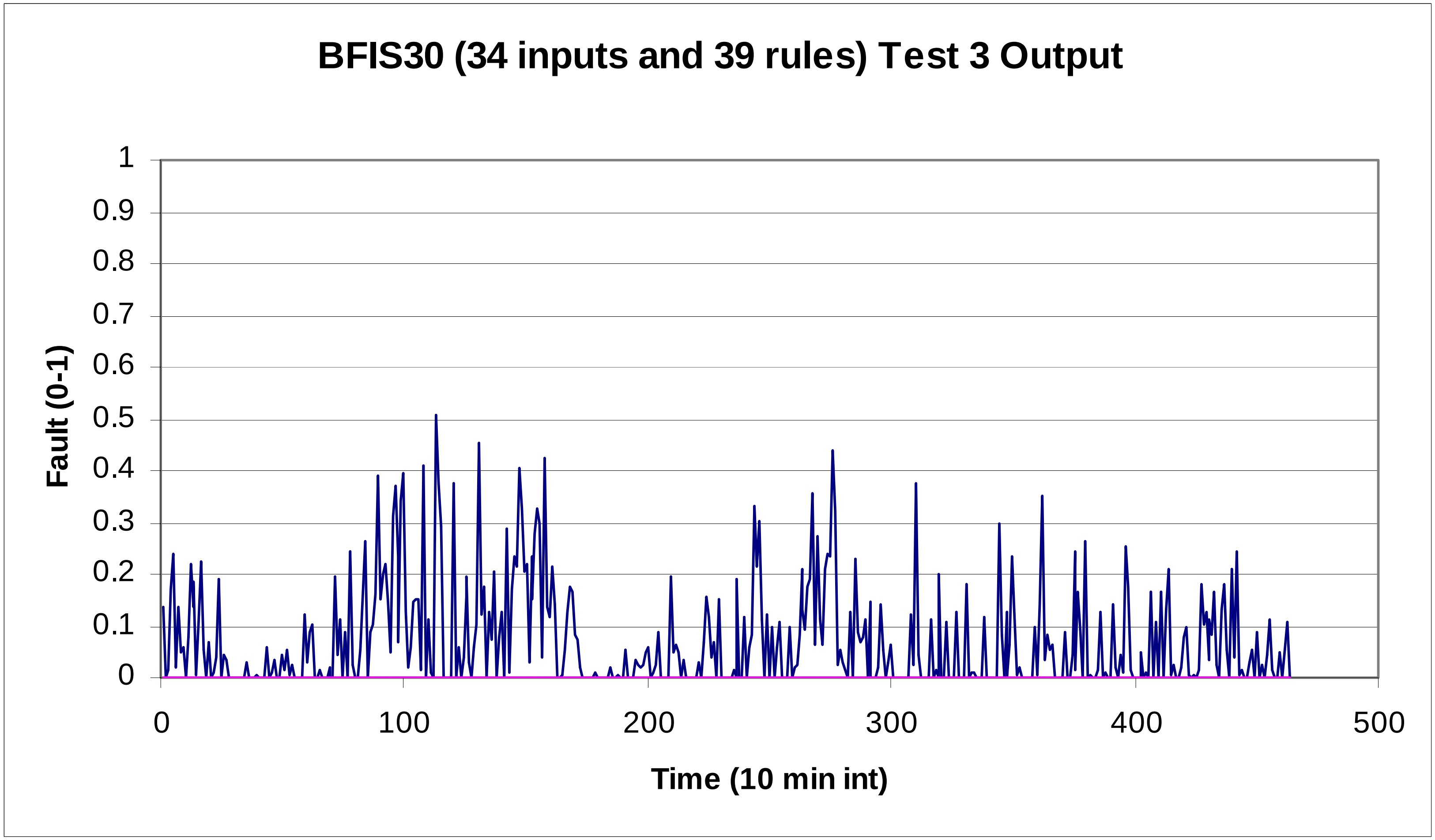

Figure 8.6 The NF Output During The Third Testing Data Set (Normal Operation)

Test 2 responses for faulty condition can be seen in Figure 8.3 and Figure 8.4. They are better for both the NN and NF compared with test 1. The responses have less noise especially for the NF FDI system. The periodic maintenance disturbance is very pronounced in the NN while it is significantly reduced for the NF response. The detection time for the NN is about 120 minutes while it is 10 minutes for the NF. Covering the leaves took about 30 minutes and the NF FDI system started identifying the problem in the end of the covering process. Correct classification is 75% for the NN and 100% for the NF.

Both FDI systems have good responses for the normal conditions as seen in Figure 8.5 and Figure 8.6. Both responses are a little noisy but most points are below the normal condition limit. Correct classifications are 98% for the NN and 100% for the NF. The summary of the NN and NF FDI systems performances can be seen in Table 8.1.

Table 8.1 The ANN and The NF FDI Systems Performances

Test # and Type	FDI system	Detection time	Misclassification	Correct classification
Test 1 Transpiration Fault	NN	180 minutes	±35%	±65%
	NF	50 minutes	0.60%	99.40%
Test 2 Transpiration Fault	NN	120 minutes	±25%	±75%
	NF	0 minutes	0%	100%
Test 3 Normal condition	NN	NA	±2%	±98%
	NF	NA	0%	100%

8.2 Mechanical Fault Responses Comparison

The NN and NF FDI responses for the normal condition test can be seen in Figure 8.7 and Figure 8.8. Both the NN and NF FDI systems can identify the normal condition very well. Correct classification is 100% for both systems.

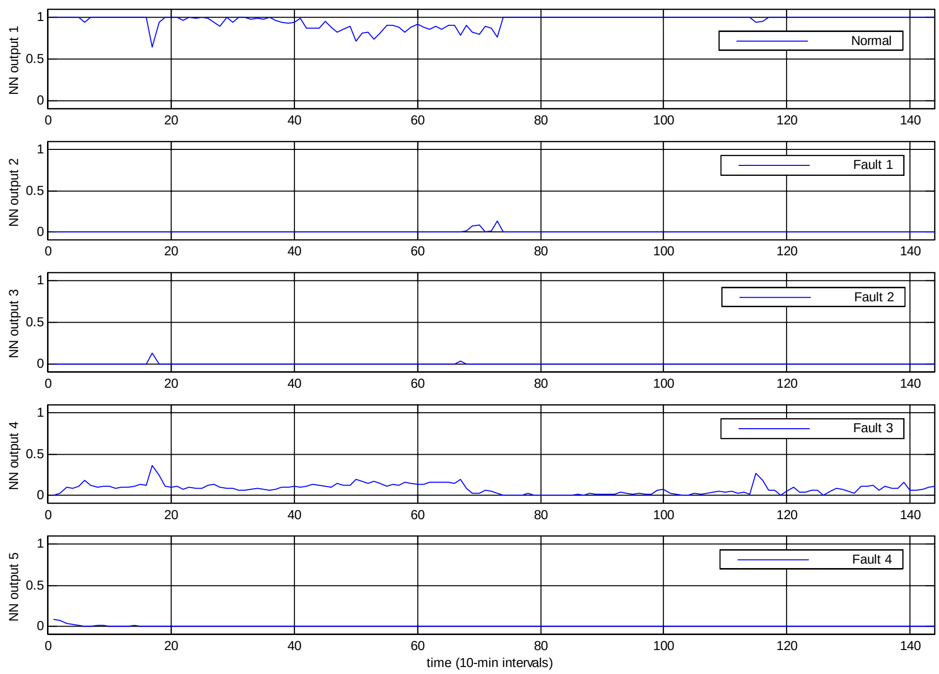

Figure 8.7. The NN (Courtesy of Ferentinos, 2002) Outputs During a Data Set of normal operation

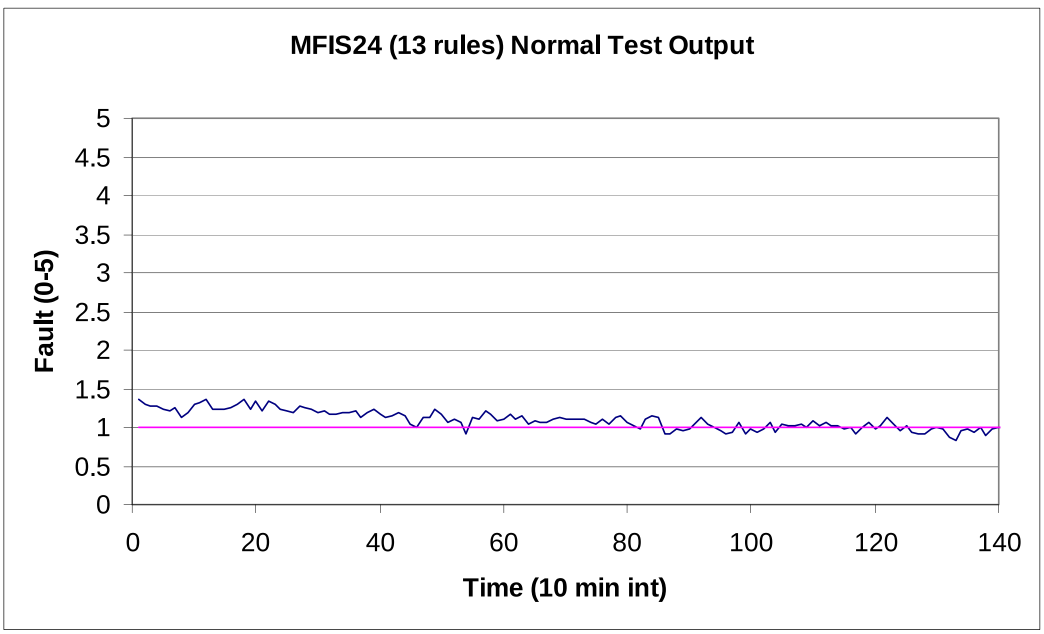

Figure 8.8. The NF Output During a Data Set of Normal Operation

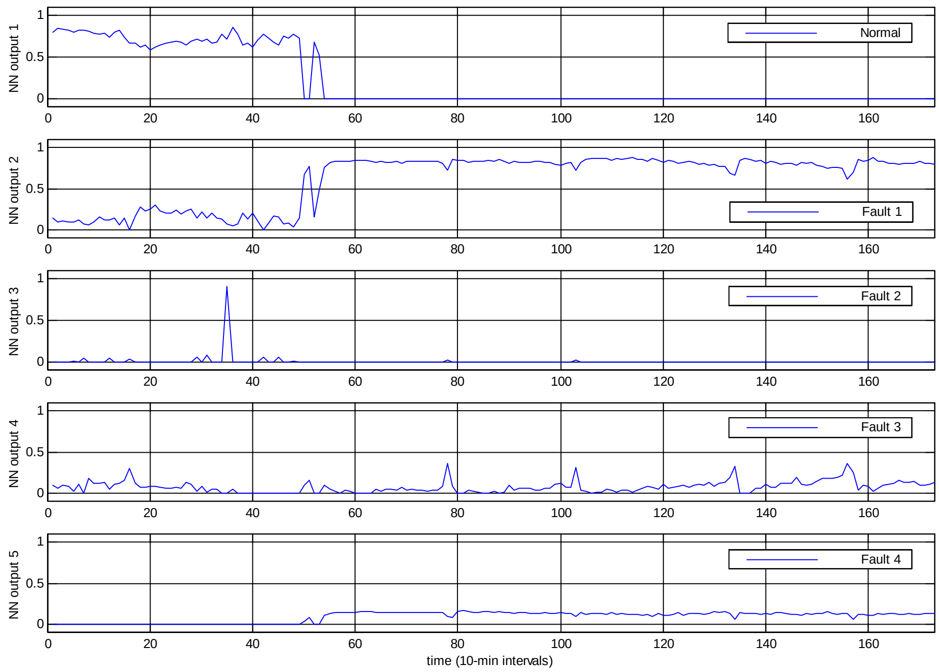

Figure 8.9 The NN (courtesy of Ferentinos, 2002) Outputs During pH Pump Fault.

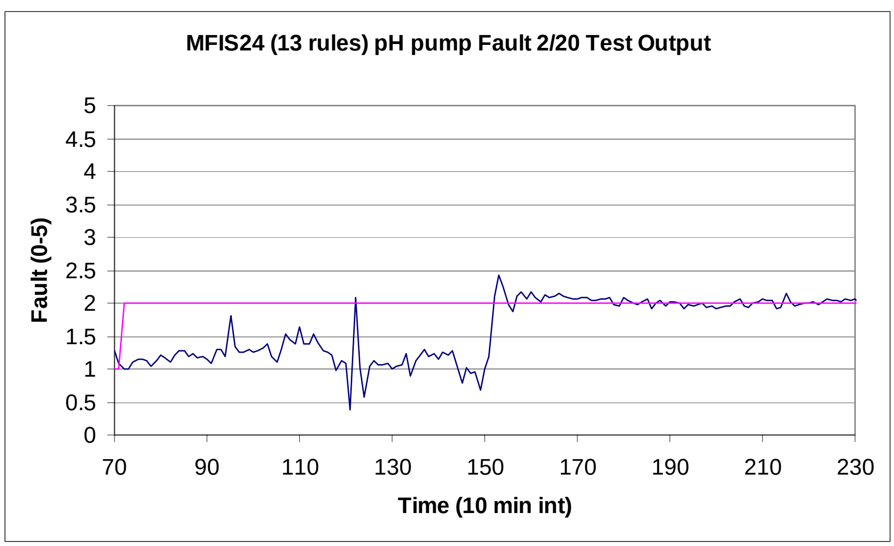

Figure 8.10. The NF Output During pH Pump Fault

Responses for “Fault 1” (the pH control pump fault) are shown in Figure 8.9 and Figure 8.10. There is a little up and down movement in the NN and NF FDI systems in the beginning of the fault (from point 0 to 50 in the NN response and from point 71 to 151 in the NF response). A possible explanation for this condition is that the FDI systems were trained to identify the condition as fault 1 where there was no symptom available until the pH control pump was asked to add acid and failed to do so. They picked up noise and identified this condition as a faulty condition. The responses of the neuro-fuzzy FDI system are mostly below 0.4 during the period without pH control except for two spikes which can be regarded as noise. Detection time for the NN is about 500 minutes and for the NF is about 800 minutes. As discussed in chapter 7, the pH control pump was not always working. If the pH value was below the set point of 5.8, the pH controller did not request additional acid from the pump. This condition could exist for 8 or more hours.

Both FDI systems identified the faulty condition immediately and steadily after the pump failed in fulfilling the pH control signal request. The output of the NN FDI system stayed at a steady state value of 0.8 during the real pH control pump fault.. The NN FDI systems may not fully recognize the symptom pattern as fault 1 and small deviations occurred at the fault 3 and 4 output. The NF output remained close to the desired fault level 2 after the pH control pump failed to inject the requested acid, starting from point 151. The faulty condition was fully recognized as a pH pump fault condition and the real detection time was about 30 minutes.

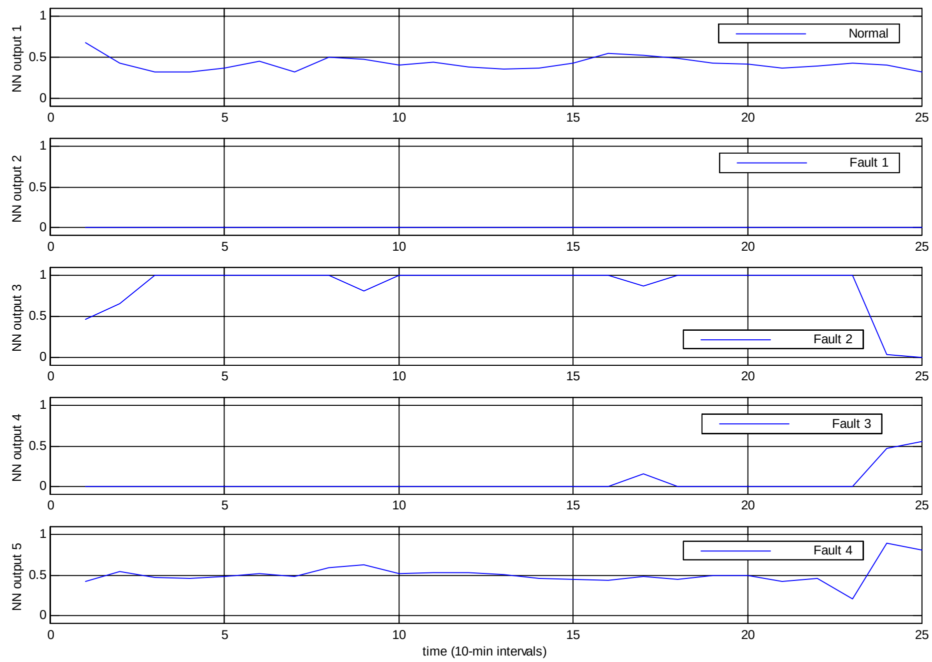

Figure 8.11. The NN (courtesy of Ferentinos, 2002) outputs during a circulation pump fault

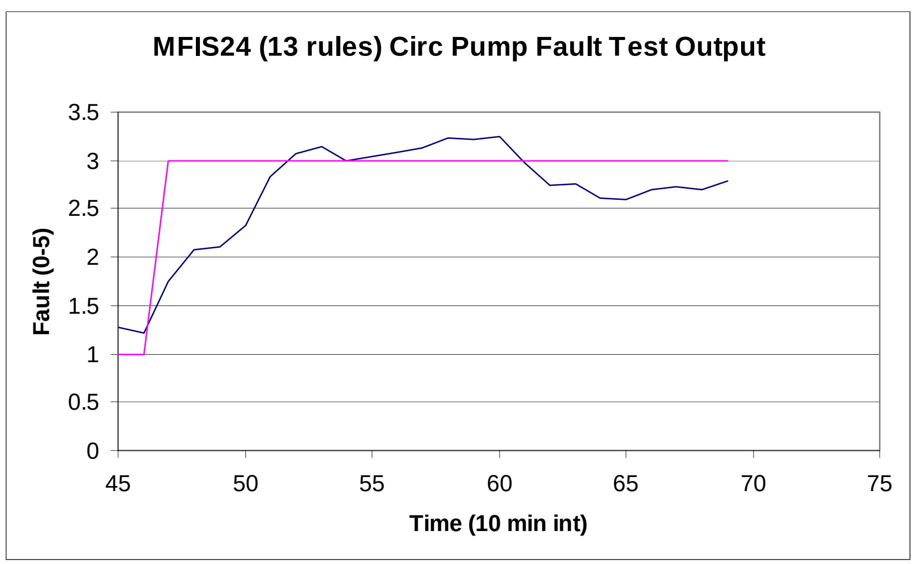

Figure 8.12. The NF Output During a Circulation Pump Fault

Circulation pump fault responses for both systems can be seen in Figure 8.11 and Figure 8.12. As can be seen from the response in Figure 8.8, this particular fault was problematic for the NN. This problem was especially pronounced near the end of the fault response. Output 1 for normal condition and output 5 for EC sensor fault condition stayed around 0.5 for most of the testing period. For the last few points, the NN FDI system gave a false identification as EC sensor fault. The NF FDI system detected the fault correctly for the whole dataset. The symptoms for the last few points were weakened but the response was still above the threshold value of 2.6.

pH sensor fault responses for both NN and NF can be seen in Figures 8.13 and 8.14. The NN FDI system can recognize the faulty condition very well except for the last few points. There are several misidentifications for circulation pump fault and EC sensor fault, all are coincident with the weakening detection for pH sensor fault. The periodic weakening of pH sensor fault detection was caused by the sinusoidal nature of the drifting simulation. Whenever the pH value came close to normal, the fault symptom weakened. Similar to the response for circulation pump fault, there were a few points at the end of experiment where the response gave false identification of the faulty condition as circulation pump fault and EC sensor fault.

The NF FDI system also had the similar periodic response compared with the NN FDI system. Whenever the sinusoidal drifting went to around normal, the system response went back to fault level 2. This behavior posts a serious problem for any NF FDI system since the up and down nature of the output value gives misclassification of the fault. As discussed in the chapter 7, using a fault value close to normal for the fault with oscillating behavior can significantly reduce this problem. If the pH sensor fault were put on level value 2, the fault value 4 in Figure 8.8 would be equal to 2. Value of 2 would be equal to 1.5, the half point between normal and the fault. In this configuration, only one point of the response is misclassified.

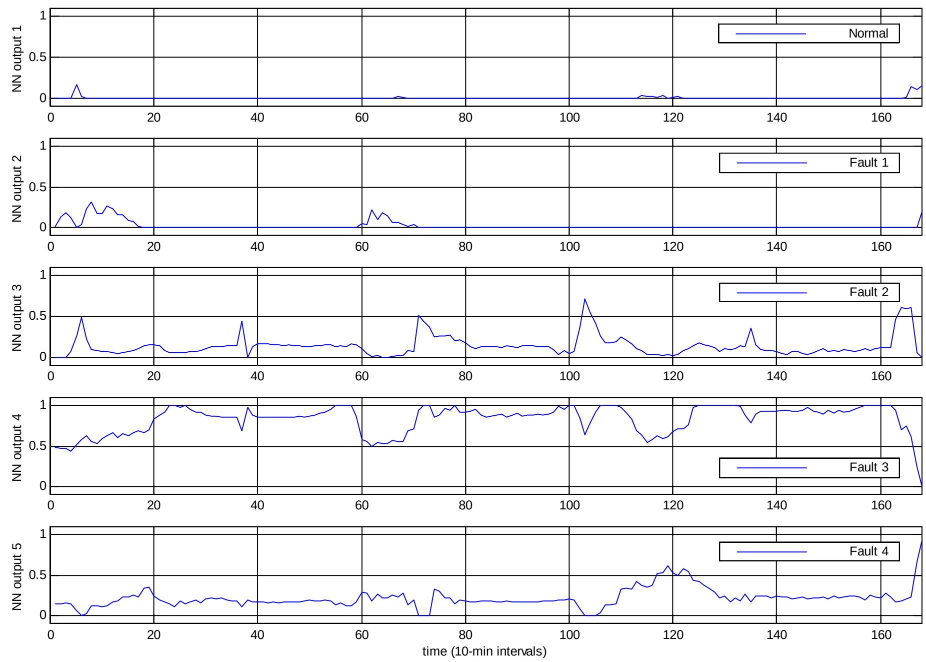

Figure 8.13 The NN (courtesy of Ferentinos, 2002) outputs for pH sensor fault

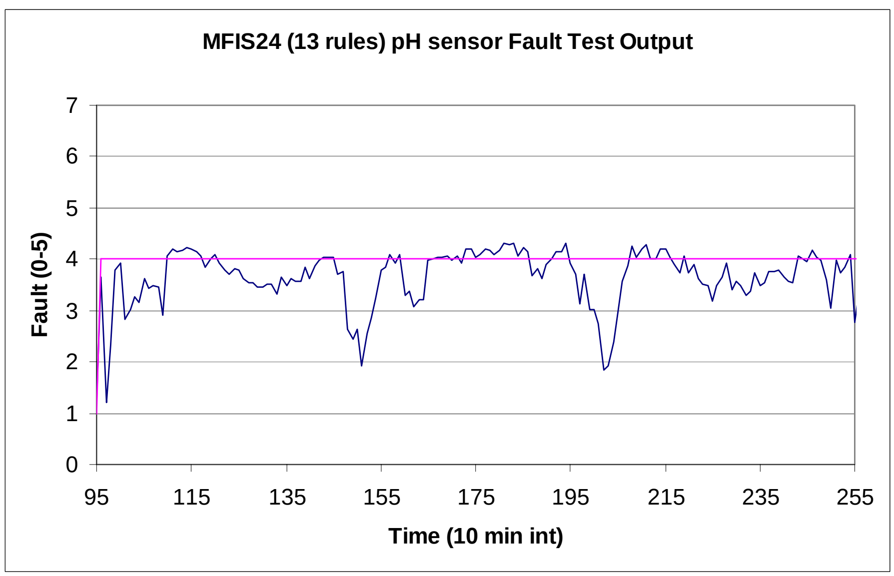

Figure 8.14. The NF output for pH sensor fault

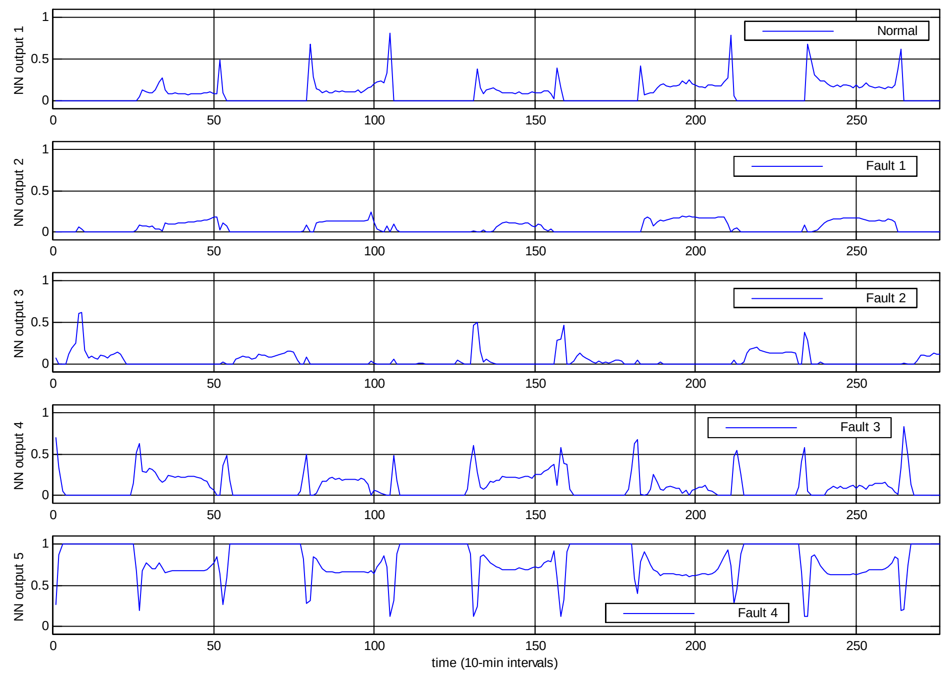

Figure 8.15 (courtesy of Ferentinos, 2002) The NN Outputs During EC Sensor Fault

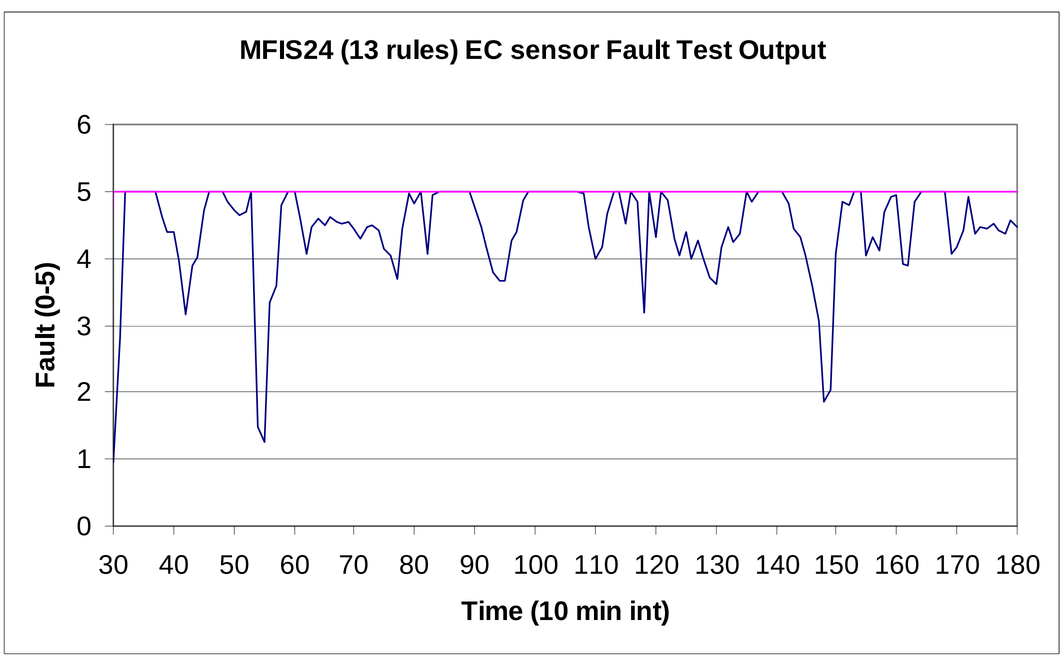

Figure 8.16 The NF Output During EC Sensor Fault

The NN and NF FDI system responses for EC sensor fault can be seen in Figures 8.15 and 8.16. The NN FDI system response has a periodic up and down pattern for “normal”, “Fault 3” (pH sensor fault) and “Fault 4” (EC sensor fault) system outputs. The FDI system responses correctly identified the “Fault 4” condition most of the time. Drifting of the electrical conductivity (EC) sensor is also represented with a sinusoidal wave so every time the EC value got closer to the normal value, the symptom weakened and was shown as “Normal” and “Fault3” outputs.

The response of the NF FDI system also has a periodic pattern. Similar to the NN FDI system, the pattern is repeated in about 100 points, the period of one sinusoidal wave. As in the previous fault comparison, a fault with periodic drifting should not be located too far from the normal condition since the oscillation will cause false identifications. If the EC fault were located next to normal, the fault value of 5 would be 2. With this arrangement, only two points would cross the halfway point to normal and be misclassified. The NN FDI system response would be identified correctly 98.6% of the time.Fault detection and identification system performance (as discussed before) is measured by detection time and correct classification. Table 8.2 shows detection times of the NN FDI and NF FDI systems.

The NN FDI system has faster detection time for fault 1 and 2 while the NF FDI has faster detection time for fault 3 and 4. Fault 1 detection times depend strongly on the testing data set. If the pH value stays below the set point for 8 hours, the failure of the pH pump can not be detected for at least 8 hours, so the detection time in Table 8.2 is a bit misleading for “Fault 1”. If the detection time definition for this particular fault is revised as the time between when the pH control program asks the pH pump to inject acid and the faulty condition is detected at the output of the FDI system, the detection time for the NN FDI and NF FDI system responses are about the same at 30 minutes.

Table 8.2. Detection time of the NN and the NF FDI systems

	Type	10min	20min	30min	40min	1h	1.5h	6h	8h	13h
Fault 1	NN	0%	0%	0%	0%	0%	0%	33%	66%	100%
	NF	0%	0%	0%	0%	0%	0%	0%	50%	100%
Fault 2	NN	50%	100%	100%	100%	100%	100%	100%	100%	100%
	NF	0%	100%	100%	100%	100%	100%	100%	100%	100%
Fault 3	NN	0%	0%	0%	0%	0%	50%	100%	100%	100%
	NF	0%	0%	100%	100%	100%	100%	100%	100%	100%
Fault 4	NN	0%	50%	100%	100%	100%	100%	100%	100%	100%
	NF	0%	100%	100%	100%	100%	100%	100%	100%	100%

Correct classification of the operating condition is another important measure of the FDI system performance. Table 8.3 shows the classification percentages of data samples for normal and faulty conditions. The normal condition is correctly classified by the NN and NF FDI system about 99% of the time. By looking at the response chart for both NN and NF FDI systems, "Fault 1" responses are the best among fault responses but it does not represent the number in Table 8.3. The pH control pump is not always on as it was mentioned above. If the classification definition starts at the time when the pH control pump failed to inject acid, the correct classification jumps to 99% for both NN and NF FDI system.

The false classification of 1% is only between the first time the pH control system asks the pH pump to inject acid and the time when the fault is detected at the FDI system's output. The discrepancies during this period can be classified as detection time and it is not misclassification as used by Ferentinos. Ferentinos defined correct classification for every sample points including any discrepancies during the detection time period. With this definition, systems with slower detection time will be penalized twice; they have long detection time in addition to a higher misclassification percentage.

Table 8.3. Classification percentages of data samples of normal and faulty conditions

Tested data set	Type	Classification in:					
		Normal	Fault 1	Fault 2	Fault 3	Fault 4	Unknown fault
Normal	NN	99.2 %	0.2 %	0.4 %	0.2 %	0 %	NA
	NF	99%	1%	0%	0%	0%	NA
Fault 1	NN	25.5 %	70.1 %	0.2 %	0%	4.2 %	0
	NF	43.7%	56.3%	0%	0%	0%	NA
Fault 2	NN	1.9 %	0%	92.4 %	0%	3.8 %	1.9 %
	NF	0%	17.4%	82.6%	0%	0%	NA
Fault 3	NN	0%	0%	1.5 %	92.1 %	3.9 %	2.5 %
	NF	0.3%	3.9%	13.2%	77.5%	5.1%	NA
Fault 4	NN	1.8 %	0	1.7 %	2.4 %	92.9 %	1.2 %
	NF	0.6%	1.7%	1.7%	29.4%	50.6%	NA

The NN FDI system has better classification percentage compared with the NF FDI system for the “Fault 3” and the “Fault 4” conditions. Although these responses are strong points for the NF FDI system, incorrect fault level arrangement made it look like the weakest points. By adjusting both periodic faults to level next to normal condition the correct classification percentage can jump to 99.4% for “Fault 3” and 98.6% for “Fault 4”.

CHAPTER 9

CONCLUSIONS AND FUTURE RESEARCH NEEDS

9.1 Conclusions

Using the indirect way to detect faults in this thesis is shown to be satisfactory. The NF FDI system has a very good pattern recognition and generalization capability as shown in the test results. The NF detection system can readily recognize desired faults in the plant production system.

The NF systems with up to 39 inputs were tested and shown to be satisfactory in detecting faults in the hydroponic system although some literature does not recommend using more than 14 inputs (Jung 1998). The FDI systems with more inputs performed better than systems with a lower number of inputs. The NF biological FDI system with 39 inputs (the highest number of inputs) had the least training error and the best performance in detecting a severe transpiration fault in the system. The transpiration fault was detected in 50 minutes and the misclassification was less than 1%. A simple heuristic filter discussed in Chapter 6 can improve the correct classification to 100%.

The NF biological FDI system with a dedicated output for transpiration fault performed satisfactorily. It can tolerate maintenance and harvesting period disturbances better than the ANN. It can detect the fault in less than 50 minutes, which is half the time needed by the ANN.

The multi level value fault detection system is simpler than a multi output system. The widespread method of using a single system to detect one fault and combine many systems in parallel needs even more training time. This characteristic can significantly reduce development time for implementing the FDI system for the grower’s particular system. The multi level value NF FDI system tested in this research performed satisfactorily in testing using real experimental data and can be applied directly in real time production systems. The real advantage of this system is the simplicity of the multilevel value output. The grower only has to see one real time graph with each level on the graph representing different system conditions.

If the error is the only indicator, the worst FDI test results are for the sensor faults. In the real operating condition, the drifting is only one-way and not periodic. If the drifting is only one-way, the fault detection system has less noise since it does not go back and forth from faulty condition to normal. Using these faults to represent incipient and intermittent faults at the same time really tests the capability of the FDI systems. By reducing these faults to one-way drifting, the FDI system responses to these faults are comparable to other faults.

A combination of redundancy type of fault detection with the indirect type of fault detection developed in this research produces a very robust fault detection system for the plant production system. Duplicating sensor and mechanical components make the down time for maintenance close to zero so the production system is not disturbed. Growers must consider the balance of cost and benefit for this setup. A robust and reliable system guarantees uninterrupted production.

Manual adjustment and maintenance of the production system should be reduced as much as possible. The random nature of the manual adjustment is difficult to be modeled in the FDI systems and caused much of the noise in the FDI system responses. Automatic regulation of water level and nutrient solution concentration (EC) can minimize these disturbances. In addition, these control signals can be used as additional inputs to increase FDI system sensitivity.

9.2 Future Research Needs

With rapid advancement in sensor technology, it should not take a long time to have more sensitive and reliable sensors for non-disturbing continuous control and monitoring of ion concentrations and plant's states. The FDI algorithm developed in this thesis is ready to explore more and accurate details for biological faults with these sensors.

The three experimental tanks had identical numbers of lettuce plants and nearly identical nutrient solution conditions during the experiments. Unfortunately, the evapotranspiration of the three tanks was found to be different and, as the result, the rate of the nutrient change in the solution for each tank was different. This condition presented additional noise and reduced the overall sensitivity for the FDI system. The problem was pinpointed as the airflow above the plants had different flow rates and patterns. This caused the evapotranspiration to be larger in one tank compared with the others. The detail information about the difference of the evapotranspiration between the tanks can be found in Appendix B. The result of this research is encouraging to make the condition of the experiment tanks as identical as possible so, the FDI system would have better results.

The research was based on the hypothesis that the most important fault had to be placed next to the normal condition and the least important error was positioned the farthest from the normal condition. In a real production system, early detection of significant faults that could cause complete failure should be prioritized. It is thereby reasonable to position the more dangerous fault close to normal since it would have less noise and the best sensitivity.

Since there were 5 different conditions for the multilevel value mechanical FDI system, optimizing the fault positions was important to minimize the errors and choose the best FDI system. Since the output was one dimensional, the arrangement of the fault position influenced the error for that particular fault in the FDI system. The pH control pump fault with a value of 2 had the best position. It was positioned next to normal condition with a value of 1. In detecting that fault, the FDI system output went directly from normal to that particular fault without crossing another fault. On the other hand, the EC sensor fault had a value of 5, to detect this condition the FDI system output in normal condition had to go through the other faults first before reaching this particular fault. If the symptoms for that particular value were weak for a while, the fault value would go through another fault before reaching normal. So its sensitivity to noise was multiplied by the distance and as a result had more misclassification.

The distance can be used as a quantifier for amplified noise for the higher value faults. The pH control pump fault that had a value of 2 was used as a standard since the distance from normal condition to this fault was 1. For the circulation pump fault with the fault value of 3, the distance was 2. If noise and weak symptoms of a circulation pump fault caused the fault value to oscillate between 1 and 3, the error would be twice as large as the pH control pump fault that oscillated between 1 and 2. A fault value of 4 had three times the distance of pH control pump fault and fault value of 5 had fourth time the distance.

The error calculated for the test was Root Mean Squared Error (RMSE), which gave more penalties to bigger deviations and less for small ones. RMSE is defined as the sum of all squared error divided by population size and then taken to the square root. Since it is not linear (average error is) we can't just divide the error from fault value 5 by 4 to equalize it with the error of the pH control pump fault. Average error is defined as the sum of all errors divided by the population size. Since this error is linear, it can be used directly to compute equalized error and can be seen in Table 9.1. The error value of fault level 5 was divided by 4 to equalize the distance error. Fault level 4 was divided by 3 and fault level 3 was divided by 2. Equalized distance errors can be seen in Table 9.2.

The order of the test result based on equalized error in Table 7.9 is:

1. Test 2 for pH control pump fault with the best error of **0.057871024**
2. Test 1 for normal condition with the best error of **0.088077271**
3. Test 5 for EC sensor fault with the best error of **0.113126188**
4. Test 3 for circulation pump fault with the best error of **0.128570859**
5. Test 4 for pH sensor fault with the best error of **0.131756572**

Fault value of 2 for pH pump fault has the least error of 0.057871024. This number is less than half of 0.131756572 for worst tested fault value of 4. This is a very significant difference in the errors and could mean that the arrangement of faults can be improved in the future. Since the error will be multiplied for the faults farther from normal, it is logical to put the faults with least errors farther away from normal to minimize the interference with neighboring faults.

The range of error and the average error of the responses can also be used as additional information in determining the best placement of the fault in the multi level value FDI system. If the least error of a particular fault for the FDI systems is small but the average error of all FDI systems for that fault is large then there might be a problem in putting the fault farther from normal. This means the trained FDI systems with different compositions will give a wide range of performance for this fault. The error range data will more strongly support this observation. The compromise is that if the fault responses for the tested systems are broad in range, a fault level position closer to normal is better.

Table 9.1 Mean Error Testing Result of Mechanical Fault Detection System with 5-minute interval

roi	Rules	Training Error	Testing Error: test 1	test 2	test 3	test 4	test 5	Error Sum
19 INPUTS								
0.25	14	0.561	0.125576741	**0.063033321** [b]	**0.34526135** [b]	**0.395269715** [a]	0.530722196	**1.712296323** [a]
0.3	8	0.6035	0.12698805	0.123321373	0.491569831	0.62822609	0.786531767	2.409070111
24 INPUTS								
0.22	18	0.4318	0.124162224	0.095590199	0.413854572	**0.520543229** [d]	**0.483655953** [b]	**1.890239176** [b]
0.25	13	0.4539	0.118698785	0.140794917	0.367165126	**0.505566581** [b]	**0.51908931** [d]	**1.903747719** [c]
0.28	9	0.5001	0.114925991	0.093381861	**0.349393546** [c]	0.739713801	0.582237733	2.132085932
0.3	8	0.5166	0.114623148	0.118987069	0.425160061	0.878250279	0.602420807	2.391874364
29 INPUTS								
0.234	17	0.4509	**0.088077271** [a]	**0.081255945** [c]	0.459120707	**0.517740734** [c]	**0.50738127** [c]	**1.906008927** [d]
0.24	16	0.4542	**0.088440736** [b]	**0.057871024** [a]	0.431374943	0.600352722	0.600123927	2.030596352
0.25	13	0.4699	0.1346544	**0.090017311** [d]	**0.257141718** [a]	0.869660412	0.534386353	2.138293195
34 INPUTS								
0.26	11	0.4731	0.173404449	0.14899026	0.368275121	0.541802246	**0.452504751** [a]	1.937409828
0.28	8	0.4672	**0.114269849** [d]	0.126929753	**0.358752871** [d]	0.648502741	0.55676262	2.057650835
0.3	7	0.5279	**0.110788147** [c]	0.121773645	0.569490673	0.923487155	0.848829811	2.826802431

[a] the best training result   [b] the 2nd best   [c] the 3rd best   [d] the 4th best

Table 9.2 Equalized Distance Errors for system with 5-minute interval

roi	rules	Training Error	Testing Error					Error Sum
			test 1	test 2	test 3	test 4	test 5	
				19 INPUTS				
0.25	14	0.561	0.125576741	**0.063033321** [b]	**0.172630675** [b]	**0.131756572** [a]	0.132680549	**0.625677857** [a]
0.3	8	0.6035	0.12698805	0.123321373	0.245784915	0.209408697	0.196632942	0.902135977
				24 INPUTS				
0.22	18	0.4318	0.124162224	0.095590199	0.206927286	**0.17351441** [d]	**0.120913988** [b]	**0.721108106** [b]
0.25	13	0.4539	0.118698785	0.140794917	0.183582563	**0.168522194** [b]	**0.129772328** [d]	**0.741370786** [c]
0.28	9	0.5001	0.114925991	0.093381861	**0.174696773** [c]	0.246571267	0.145559433	0.775135325
0.3	8	0.5166	0.114623148	0.118987069	0.21258003	0.292750093	0.150605202	0.889545542
				29 INPUTS				
0.234	17	0.4509	**0.088077271** [a]	**0.081255945** [c]	0.229560353	**0.172580245** [c]	**0.126845318** [c]	**0.698319131** [d]
0.24	16	0.4542	**0.088440736** [b]	**0.057871024** [a]	0.215687472	0.200117574	0.150030982	0.712147788
0.25	13	0.4699	0.1346544	**0.090017311** [d]	**0.128570859** [a]	0.289886804	0.133596588	0.776725963
				34 INPUTS				
0.26	11	0.4731	0.173404449	0.14899026	0.184137561	0.180600749	**0.113126188** [a]	0.800259207
0.28	8	0.4672	**0.114269849** [d]	0.126929753	**0.179376436** [d]	0.21616758	0.139190655	0.775934274
0.3	7	0.5279	**0.110788147** [c]	0.121773645	0.284745336	0.307829052	0.212207453	1.037343633

[a] the best training result [b] the 2^{nd} best [c] the 3^{rd} best [d] the 4^{th} best

Table 9.3 Least, Most, Average and Range of Errors for Systems with 5 minute interval

	test 1	test 2	test 3	test 4	test 5
Least Error	**0.088077271**	**0.057871024**	**0.128570859**	**0.131756572**	**0.113126188**
Most Error	0.173404449	0.126929753	0.284745336	0.307829052	0.212207453
Average Error	0.119550816	0.105162223	0.201523355	0.21580877	0.145930135
Range	0.085327178	0.069058729	0.156174477	0.17607248	0.099081265

The range and the average error of the FDI5 system responses are shown in Table 9.3. Test 4 had the most error so this fault should be positioned as close as possible to normal to avoid multiplication of error and reduce the chance of misclassification. The next worst fault was circulation pump fault. The third was the EC sensor fault and the last was the pH control pump fault.

The order of the fault should be:

- Fault value 1 for **Normal**
- Fault value 2 for pH sensor fault
- Fault value 3 for circulation pump fault
- Fault value 4 for EC sensor fault and
- Fault value 5 for pH control pump fault

This fault arrangement is good but the result can be improved with a little modification of the fault position. By positioning the normal condition in the middle of the multi level fault system, the distance the response had to go through to reach the fault was minimized. Two of the faults with least errors can be placed at the farthest position from normal. The next two can be placed at the next two farthest positions and so on. In this research the pH control pump and EC sensor faults had the least errors so these could be positioned as the outer faults. The circulation pump and pH sensor faults can be positioned as the inside faults.

The order can now be arrange as:

- Fault Value 1: pH control pump fault
- Fault Value 2: Circulation pump fault
- Fault Value 3: **Normal**
- Fault Value 4: pH Sensor fault
- Fault Value 5: EC Sensor fault

With this arrangement, the greatest distance the response had to travel to go to the outer faults was 2. The pH control pump fault and circulation pump fault shared similar characteristics so it was ideal to position them at the same side of normal. In this way, if the symptoms for that fault were weak, the response did not have to go through the other side and caused misclassification.

APPENDIX A

SENSORS AND EQUIPMENTS

A-1 Greenhouse

Experiment Location was at greenhouse #15 section D, Kenneth Post Laboratory, Cornell University, Ithaca, NY 14853, USA with latitude of 42.440N and longitude of 76.496W and elevation about 1100 feet from sea level. This particular greenhouse had 5 identical sections (A-E) which each section had a floor area of 85 m^2. The greenhouse #15 stretched from east to west direction.

The environment of these five sections was controlled by a central computer via Analog Device's 6B micro controller module in each of the greenhouse section.

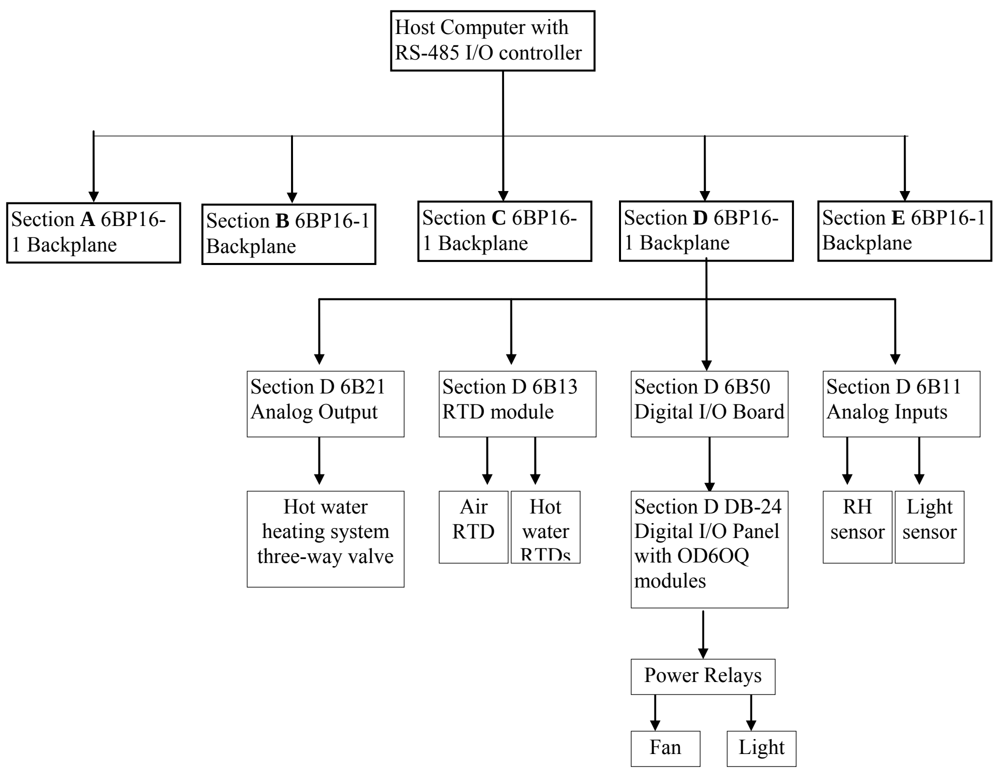

Figure A-1. Greenhouse control and monitoring system

Hardware list:

- 6B16-1 Backplane connected to the PC host using RS-485 connection
- 6B13 Temperature Input Module for air temperature and hot water temperature reading from RTD sensors
- Platinum RTD (Resistance Temperature Device) Temperature Sensors
- 6B11 General Purpose Analog Input Module for LI-Cor Quantum Light, relative humidity and CO_2 sensor reading
- 6B21 Analog Output Module for controlling hot water three way valve
- Johnson Controls three-way valve with 4-20 mA input
- 6B50 Digital I/O Board with 4-6 VDC output
- DB24 based OD6OQ with 3-60 VDC outputs for relay driver
- Omron G3NA-255B power relays for lights, fans, CO_2 and shade control

A-2 Hydroponic System

Hardware List:

- PCI-MIO-16xe-50 20kS/s, 16 bit,16 Analog Input, 2 Analog Output,, 8 Digital I/O
- SCXI 1001 12 slot SCXI chassis
- SCXI-1124 6 channel isolated DAC module
- PHCN-420 pH controllers with 4-20 mA input from Omega Engineering Co.
- SCXI-1161 8 channel power relay
- O2 supply Tank
- ASCO model 8016G Red Hat II ignition proof solenoid valves for O_2 supply
- SCXI-1122 16 channel isolation input amplifier with excitation
- PR-11 Platinum RTD Temperature probes from Omega Engineering Co.

- PHE-900 HF-Resistant Alpha pH electrodes from Omega Engineering Co.
- PHP-75-MA Chemical Metering Pumps
- CDCN-108 Non contact Conductivity sensor from Omega Engineering Inc.
- CDCN-672 Conductivity analyzers from Omega Inc.
- DO controller 1000 ¼ DIN Dissolved Oxygen Controller from Cole- Palmer Instrument Co.
- OAKTON 35640-50 Industrial Dissolved Oxygen Probe from Cole-Palmer Instrument Co.
- Hach Corporation APA 6000 Nitrate/ Ammonium Process Analyzer.
- SP 652-A5-250Kg-1MYY single point Scale and BT84 Digital Scale Indicators from B-TEK Inc.
- Grainger Submersible pumps model 1P808 for nutrient solution circulation

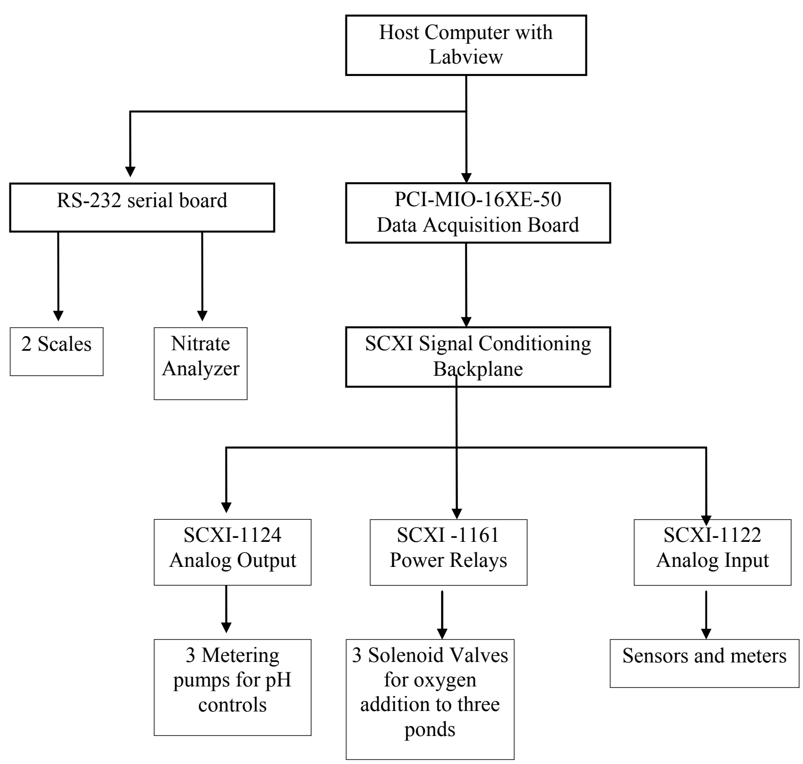

Figure A-2. Hydroponic system control and monitoring

APPENDIX B

EVAPOTRANSPIRATION

B-1 Evapotranspiration Difference Between Tanks

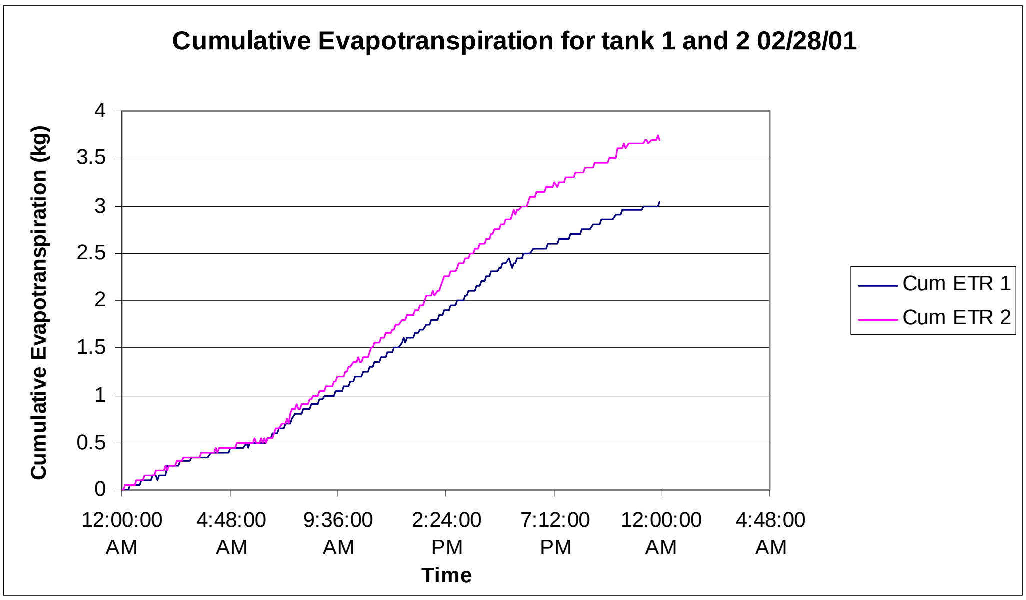

Figure B-1 Cumulative Evapotranspiration for tank 1 and 2 02/28/01

Charts of cumulative evapotranspiration for tanks 1 and 2 are shown in Figure 7.3, 7.4 and 7.5. In 02/28/01 tank 1 lost about 3.05 kg of water while tank 2 lost about 3.7 kg, a difference of 0.65 kg or about 20% of tank 1's water loss. About a week later in 03/08/01, tank 1 lost about 1.8 kg of water while tank 2 lost about 3.5 kg. This time the different is very significant at about twice the water loss in tank1.

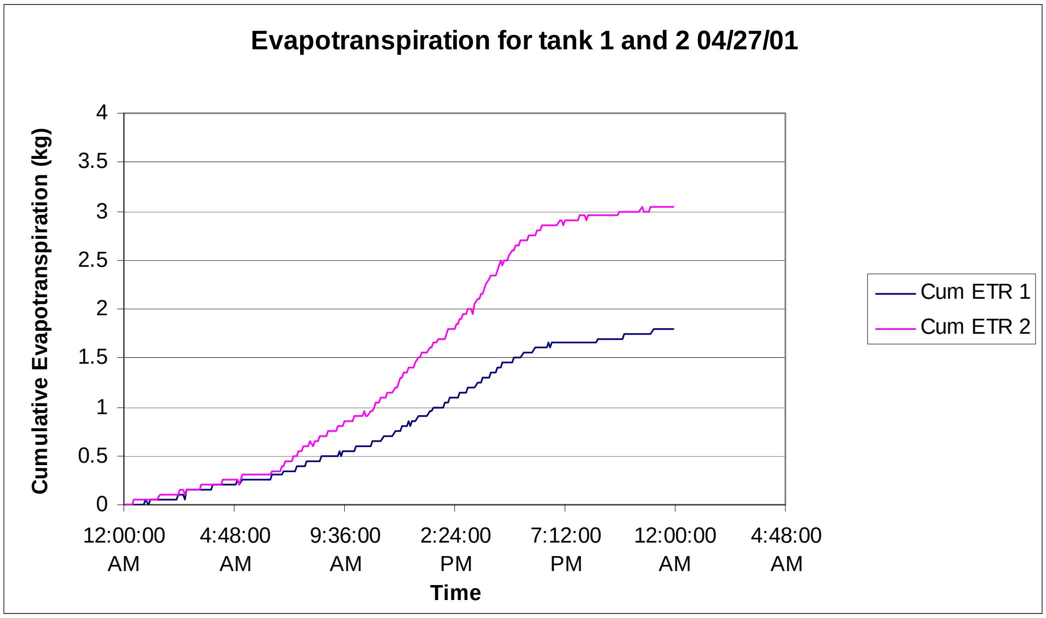

Figure B-2 Cumulative Evapotranspiration for tank 1 and 2 03/08/01

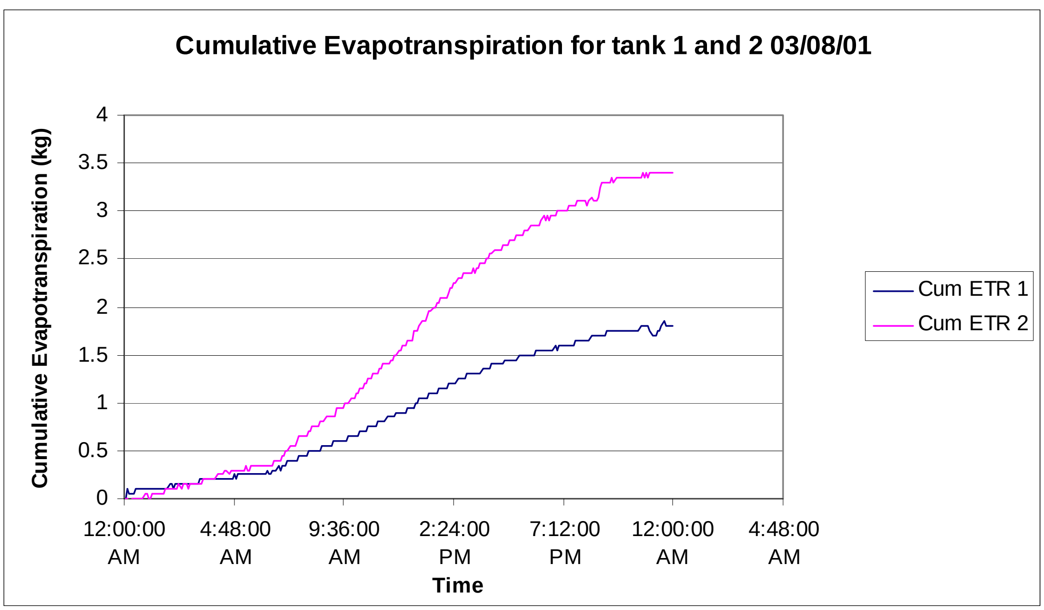

Figure B-3 Cumulative Evapotranspiration for tank 1 and 2 04/27/01

Two months after in 04/27/01, water loss is 1.8 kg and 3.05 for tank 1 and 2 respectively. Tank 2 generally had higher water loss that tank 1 although the difference varies.

APPENDIX C

MATLAB CODES

C-1 Data Standardization

```
clear
load TF5m220test.csv
load BFTrain5m.csv
x = TF5m220test;
y = BFTrain5m;
m=size(y,1);
% m is the number of rows
n=size(y,2);
% n is the number of column
p=size(x,1);
q=size(x,2);
y1=mean(y)
ystd=std(y)
for a=1:p
   for b=1:q
     xnormal(a,b)=(x(a,b)-y1(b))/ystd(b);
   end
end
xnormal(1:142,q)=-0.908277;
xnormal(143:p,q)=1.100925;
save TF5m220testn.txt xnormal -ASCII
```

C-2 ANFIS Training Program

```
clear

load bf5mwrate39input01trainn.txt
z=readfis('bf5m39inputsc025trained')

y = bf5mwrate39input01trainn;

[mf5m44444t,trainerror,stepsize] = ...
    anfis(y,z,[500 0 0.08 0.9 1.1],[])

writefis(mf5m44444t,'bf5m39inputsc025trained1')

save bf5m39inputsc025result1
```

C-3 ANFIS Test Program

```
clear

fismat= readfis('bf5m39inputsc05trained4.fis');

load bf5mwrate39inputNtestn.txt
load bf5mwrate39inputNtest.csv
load bf5mwrate39input01train.csv
x = bf5mwrate39inputNtestn;
y = bf5mwrate39input01train;
z = bf5mwrate39inputNtest;
m=size(y,1);
% m is the number of rows
n=size(y,2);
% n is the number of column
p=size(x,1);
q=size(x,2);
y1=mean(y)
ystd=std(y)
r=size(z,1);
s=size(z,2);

testinput=x(:,1:(n-1));
output1=evalfis(testinput,fismat);

testoutput=z(:,s);

for b=1:p
     xorigin1(b)=(output1(b)*ystd(n))+y1(n);
end

xorigin=xorigin1';
save bf5mwrate39inputNtestsc05fisoutput5epochs.txt xorigin -ASCII
plot(1:p,testoutput,1:p,xorigin);
```

C-4 Noise and Fault Decision Program

```
clear
load DBF10m25wratesc033t1fisoutput5epochs.txt
x = DBF10m25wratesc033t1fisoutput5epochs;
m=size(x,1);
for a=1:2
    x(a)=0
    fault(a)=0
    noise(a)=0
end
```

```
for a=3:m
    if x(a)>1
       x(a)=1;
    end
    if x(a)<0
        x(a)=0;
    end
    switch fault(a-1)
        case 0
            if x(a)<0.6
                x(a)=x(a);
                fault(a)=0;
                noise(a)=0;
            else
                counter=1;
                k=m-a
                if k<4
                    noise(a)=1;
                    fault(a)=1;
                    if k==0
                        x(a)=(x(a-1)+x(a-2))/2;
                    else
                        x(a)= (x(a-1)+x(a+1))/2;
                    end
                else
                    for b=1:4
                        if x(a+b)>0.6
                            counter=counter+1;
                        end
                    end
                    if counter >2
                        noise(a)=0
                        fault(a)=1
                    else
                        noise(a)=1;
                        fault(a)=0;
                        for c=0:4
                            x(a+c)=(x(a+c-1)+x(a+c+1))/2;
                        end
                    end
                end
             end
        case 1
               if x(a)>0.4
                   x(a)=x(a);
                   fault(a)=1;
                   noise(a)=0;
               else
                    counter= 1;
                    k=m-a;
                    if k<4
                        noise(a)=1;
                        fault(a)=1;
                        if k==0
```

```
                        x(a)=(x(a-1)+x(a-2))/2;
                    else
                        x(a)= (x(a-1)+x(a+1))/2;
                    end
                else
                    for b=1:4
                        if x(a+b)<0.4
                            counter=counter+1;
                        end
                    end

                    if counter>2
                        noise(a)=0;
                        fault(a)=0;
                    else
                        noise(a)=1;
                        fault(a)=1;
                        for c=0:4
                            x(a+c)=(x(a+c-1)+x(a+c+1))/2;
                        end
                    end

                end
           end
    end
end
noise1=noise';
fault1=fault';
save dbf10m25wratesc033filteredoutput.txt x -ASCII
save dbf10m25wratesc033noiseoutput.txt noise1 -ASCII
save dbf10m25wratesc033faultdecision.txt fault1 -ASCII



% this script file can be use for algorithm
% determining the noise and faulty condition
%if the script is used for finite data file
% and there is a possibility for a noise and
% the noise is at the end of the data file
% the script m file should be modified
% still thinking for modification for this case
% for on line noise and decision making program,
% must include delay for

% gathering the next 4 data point
```

APPENDIX D

LIST OF ABBREVIATIONS AND SYMBOLS

D-1 Abbreviations

ANFIS *Adaptive Neuro-Fuzzy Inference System*

ANN *Artificial Neural Network*

BFIS30/25 *Biological Fault Fuzzy Inference System with 30/25 inputs*

DO *Dissolved Oxygen*

EC *Electrical Conductivity*

FDI *Fault Detection and Identification*

FIS5 *Fuzzy Inference System with 5-minute interval data*

FIS10 *Fuzzy Inference System with 10-minute interval data*

HPS *High Pressure Sodium*

MLP *Multi Layer Perceptron*

NFS *Neuro-fuzzy System*

NN *Neural Network*

PAR *Photosynthetically Active Radiation*

PDF *Pseudo Derivative Feedback*

SLP *Single Layer Perceptron*

TS *Takagi Sugeno fuzzy model*

D-2 Symbols

m used as summation limit eg for i = 1 to m

μ represents the membership function value between 0 to 1 of a fuzzy set

σ is used as the one parameter of Gaussian membership that influence the width of the function

REFERENCES

Bocaniala, C.D. and Palade, V. 2006. Computational intelligence methodologies in fault diagnosis: review and state of the art in Computational Intelligence in Fault Diagnosis, Palade, V., Bocaniala, C.D. and Jain, L.C. eds. Springer-Verlag, London.

Calado, J.M.G., Korbicz, J., Patan, K., Patto, R.J., Sa da Costa, J.M.G. 2001. Soft computing approaches to fault diagnosis for dynamic systems. European Journal of Control 7:248-286.

Caprigiole, D., Liguori, C., Pianese, C. and Pietrosanto, A. 2003. On-line sensor fault detection, isolation and accommodation in automotive engines. IEEE Transactions on Instrumentation and Measurement, vol. 52, No.4, pp 1182-1189.

Chiu, S.L. 1994. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3).

Ferentinos, K. 2002. Neural network fault detection and diagnosis in deep-trough hydroponic systems. PhD Dissertation, Agricultural and Biological Engineering Department, Cornell University, Ithaca, NY.

Ferentinos, K.P. and L.D. Albright, 2003. Fault detection and diagnosis in deep-trough hydroponics using intelligent computational tools. ***Biosystems Engineering***, vol. 84(1), pp. 13-30

Jang, J.S.R. 1993. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, 665-685.

Jang, J.S.R., Sun, C.T., and Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, Prentice Hall, NJ.

Koppen-Seliger, B. and Frank, P.M. 1999. Fuzzy logic and neural networks in fault detection in Fusion of neural networks, fuzzy sets, and genetic algorithms. Jain, L.C. and Martin, N.M. editors. CRC Press Boca Raton, Florida.

Mendoca, L., Sousa, J., Sa da Costa, J.M.G.. 2006. Fault detection and isolation of industrial processes using optimized fuzzy models in Computational Intelligence in Fault Diagnosis. Springer-Verlag, London.

Nguyen, H.T. and Walker, E.A. 2000. A first course in fuzzy logic. 2nd ed. Chapman and Hall/CRC.

Pedrycz, W. and Gomide, F. 1998. An introduction to fuzzy sets: *analysis and design.* MIT Press, Cambridge, Massachusetts.

Pfeufer, T. and Ayoubi, M. 1997.Application of a hybrid neuro-fuzzy system to the fault diagnosis of an automotive electromechanical actiator. Fuzzy Sets and Systems 89 (1997) 351-360.

Phelan, R.M. (1977). Automatic Control Systems, Cornell University Press, Ithaca, NY.

Polycarpou, M.M. and Helmicki, A.J. 1995. Automated fault detection and accommodation: a learning systems approach. *IEEE Transactions on Systems, Management and Cybernetics,* SMC-25, 1447-1458.

Setiawan, A., Albright, L.D., Phelan, R.M. Application of pseudo-derivative-feedback algorithm in greenhouse temperature control. *Computers and Electronics in Agriculture*, vol. 26(3), pp. 283-302.

Shukri, M., Khalid, M., Yusuf, R. and Shafawi, M. 2004. Induction machine diagnostic using adaptive neuro-fuzzy inferencing system. In *Kowledge-Based Intelligent Information and Engineering Systems, 8th International Conference, KES 2004, Wellington, New Zealand, Sept. 20-25, 2004, Proceedings, Part III.* Springer-Verlag Berlin Heidelberg.

Simani, S. Fantuzzi, C., and Patton, R.J. 2003. Model-based fault diagnosis in dynamic systems using identification techniques. Springer-Verlag, London, Britain.

Sorsa, T., Koivo, H.N., Koivisto, H. Neural networks in process fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, vol21. no.4:815-825

Takagi, T and Sugeno, M. 1985. Fuzzy Identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 1:116-132.

Vemuri, A.T. and Polycarpou, M.M. 1997. Robust nonlinear fault diagnosis in input-output systems, *International Journal of Control,* 68, 3433-3460.

Yager, R.R. and Filev, D.P. 1994. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics*, 24:1279-1284.

Zadeh, L.A. 1965. Fuzzy Sets. *Information and Control* 8:338-353.

Zadeh, L.A. 1973. Outline of a new approach to the analysis of complex systems and decision process. *IEEE Transactions* SMC-3, 1; 28-44

Zadeh, L.A. 1994. Fuzzy logic, neural networks, and soft computing. Communications of the ACM. Vol 37(3): 77-84.