Process Fault Detection and Diagnosis Using Boolean Representation on Fatty Acid Fractionation Column

M. R. Othman[1] M. W. Ali[2] M. Z. Kamsah[3]

[1,2,3]*Department of Chemical Engineering*
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Tel: +06-07- 553 5602, e-mail: m.w.ali@fkkksa.utm.my

Abstract

Nowadays, detecting and diagnosing process fault is an important issue because it can improve system availability and protect chemical plant from accidents. There are many method introduced to conduct process fault detection and diagnosis (PFD&D), but this paper will focus on the use of artificial neural network (ANN) in detecting and diagnosing faults. ANN has the capability of recognizing multivariable pattern very well. This advantage is useful in systematically detect failures in process plant. Therefore, an algorithm for the development of process fault detection system in dynamic processes using artificial neural network (ANN) is presented. The algorithm utilizes process simulator to develop plant model in order to conduct sensitivity analysis and provide dynamic data on selected fault. Various process conditions are specified and simulated using commercial process simulator. Sensitivity analysis is conducted to identify whether or not the specified process condition effect the operation of the plant. If it does, each of the faults identified is represented by a specific Boolean representation. In other words, each fault has its own pattern indicated by a Boolean representation. Input for the ANN model will be the faulty data for all of the identified fault and the output will be the specified Boolean representation for each fault. The topology of the ANN model was founded on multilayer feed forward network architecture and the training scheme conducted using back propagation algorithm. The effectiveness of the proposed fault detection system on a simulated fatty acid fractionation column is presented. Through the proposed algorithm, various faults could be simulated and detected using the system. Results show that the system was successful in recognizing and detecting selected fault introduced within the process plant model.

Keywords:

Fault Detection and Diagnosis, Artificial Neural Network, Palm Oil Fractionation Process

Introduction

Nowadays, plant operation is becoming more complex as plants are often operated at extremes of pressures and temperatures to achieve optimal performance, making them more vulnerable to equipment and instrumental failures and deviation in process that may lead to catastrophic accidents. Industrial statistics show that although major catastrophe may be infrequent but minor accidents are very common. Results from these minor accidents, costing the society billions of dollars every year [1]. Thus, to prevent this from happening a quick detection and diagnosis of failures is needed. Human alone with their limited capability are out of hand when handling these problems. Hence automated process fault detection and diagnosis (PFD&D) can provide a good solution to a better safety in chemical process plant.

In recent years, many researches have been conducted on process fault detection and diagnosis using various methods such as knowledge based expert system (KBES), mathematical modeling and artificial neural network (ANN). Using KBES in PFD&D has the advantage where it can offer insight into problem solving in chemical plant. But it has its own limitations such as tedious nature of knowledge acquisition, the inability of the system to learn or dynamically improve its performance and the unpredictability of the system outside its domain of expertise. A knowledge-based fault detection and diagnosis was introduced through combination of heuristic knowledge (knowledge rules of operator) and procedural knowledge (mathematical models, Kalman filter algorithms and signal processing procedures) [2]. The technique however could be very complex when dialing with a nonlinear process. Because of these limitations, artificial neural network can give a better solution to PFD&D because of its usefulness in representing input-output data, making predictions in time classifying data and recognizing patterns. With these advantages fault detection and diagnosis is a promising area for the application of artificial neural network (ANN) in industry [3].

In this paper, we present the application of ANN in detecting and diagnosing fault occurred on chemical plant using Boolean representation. The earlier part of the paper discusses brief introduction to ANN. Then the framework for the development of the proposed fault detection system is presented. To implement the system, a fatty acid fractionation column is used as the study case. Final part of the paper shows the result, discussion and conclusion obtain from the work.

A Brief Introduction to ANN

The term artificial neural network (ANN) resulted from research by neurologist and artificial intelligence (AI) researchers to understand and model the behavior of neurons found in brain. From the research they proposed a highly interconnected neurons or nodes that mathematically interact with each other in a fashion unknown by the user to drawn output that maps the input-output pattern. ANN can be viewed as nonlinear empirical models that are especially useful in representing input-output data, making prediction in time, classifying data and recognizing patterns. Figure 1 shows the structure of a single node.

A node receives input from other nodes or from other sources and each input is weighted according to the value $w_{i,j}$ which called weight. If the weight is positive, it will excite the node, and increase the activation. Conversely, if the weight is negative, it will inhibit the node thus decrease the activation. The weighted signals to the node is then summed and subtracted with the internal threshold, T_i and the resulting signal, called *activation*, *h*, is sent to a *transfer function*, *g*, which can be any mathematical function; e.g. sigmoid transfer function.

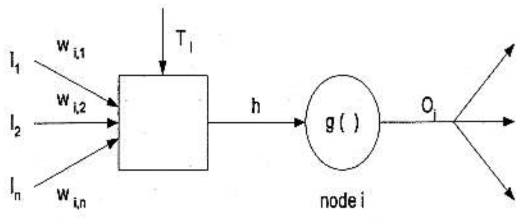

Figure 1 - Structure of single processing node [4]

When collection of nodes is interconnected with each other, as in Figure 2 it forms artificial neural network. Input layer receives information from external source and passes this information into the ANN for processing. The hidden layer receives information from input layer and quietly does all the information processing. The entire processing step is hidden from view and it may contain numbers of layers. The output layer processed information received from ANN and sends out to an external receptor.

The topology or architecture of an ANN refers to how the nodes are interconnected. These include organizing the nodes into layers, connecting them and weighting the interconnections. There are various ways nodes are connected such as interlayer, intralayer and recurrent. For interlayer connections, there are connected either by feedback or feedforward scheme.

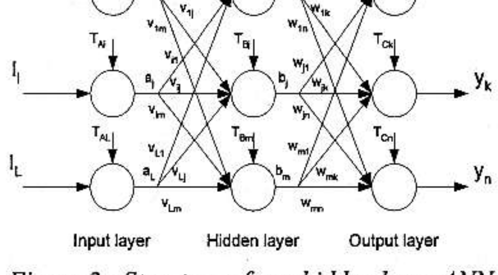

Figure 2 - Structure of one hidden layer ANN

The topology or architecture of an ANN refers to how the nodes are interconnected. These include organizing the nodes into layers, connecting them and weighting the interconnections. There are various ways nodes are connected such as interlayer, intralayer and recurrent. For interlayer connections, there are connected either by feedback or feedforward scheme. The criteria of ANN topology depends on the type of problem we are trying to solve. Basically it requires three phases to operate an ANN. The first phase is the most important and crucial part in developing ANN that is the training phase. Training phase is followed by the recall phase where the ANN is introduced to a wide array of input patterns used in training and adjustment is also introduced to make the system more reliable and robust. For the final phase, ANN is subjected to a novel input pattern for the system to perform properly.

For conducting training, backpropagation algorithm is the most investigated supervised learning algorithm [5]. There are many types of backpropagation algorithm developed and one of them is the Generalized Delta Rule (GDR) Algorithm. GDR is an iterative gradient-descent method that maps the given input with the desired output by minimizing the mean square errors, *E* using equation:

$$E = \sum_{a=1}^{A} \sum_{i=1}^{B} (t_i^a - y_i^a)^2 \tag{1}$$

where A is the number of training patterns to the input layer, B is the number of node on the output layer, t_i^a is the desired output value from the i^{th} output element given the A pattern and y_i^a is the actual output from the same node. Given the A-th pattern, the weight updating in the supervised learning algorithm is made according to the general equation:

$$v_{ij}^a = v_{ij}^{a-1} + \Delta v_{ij}^a \tag{2}$$

where v_{ij}^m is the connection weight between the i-th element in the upper layer and j-th element in the lower layer, for a-th training iteration. The changes in weight are calculated using equation:

$$\Delta v_{ij}^a = \eta \delta_j^a a_i^a + \alpha \Delta v_{ij}^{a-1} \tag{3}$$

where η is the learning rate and α is a coefficient of momentum to speed up the training rate. To calculate δ_j^a, the gradient descent for the j-th element, use the equation:

$$\delta_j^a = \left(t_j^a - y_j^a\right) \frac{\partial f}{\partial x_j} \tag{4}$$

where f is the transfer function; e.g sigmoid function and x_j is the sum of the weighted input to the j-th element for the a-th training pattern. If j belongs to the hidden layer:

$$\delta_j^a = \left(\sum_i \delta_{li}^a v_{ji}^a \right) \frac{\partial f}{\partial x_j} \qquad (5)$$

where δ_{li}^a is the gradient descent for output layer. The subscript i denotes an element in the output layer. And x_j is defined by:

$$x_j^a = \sum_i v_{ij}^a a_i^a + T_{ij}^a \qquad (6)$$

GDR calculates an error for each element in the output and hidden layers using equation (5) and (6) and recursively updates the weights of all layers using equation (3), starting from the output layer and working backwards to the input layer.

Methodology

The methodology to develop the fault detection system is described here. Figure 3 shows the overall framework of the proposed system. The plant model is developed using commercial process simulator to obtain plant experience through simulation. By using the process simulator, sensitivity analysis on the plant could be conducted as to see the effect on the plant by imposing changes to certain parameters. It also can be used to simulate and provide dynamic data on normal and faulty conditions that is not possible to obtain by using only actual plant data. The dynamic data obtain through the simulation will be used in the development of ANN model. Boolean representation developer is used to developed Boolean output vector for each of the fault the defined.

Figure 4 illustrates the algorithm for the development of the fault detection system. To obtain a plant experience a plant model that mimics the actual plant operation is developed. The next step is to select the input variables to be used in the neural network. The inputs were the measurement sensors and they are selected based on their availability and effects on the monitored variables. Thereafter, process operation and various process conditions are defined and simulated. Sensitivity analysis is conducted on each condition to determine whether it effects the operational of the plant. If it does, the condition is considered as a fault and for each of the identified fault a specific Boolean representation is defined and created.

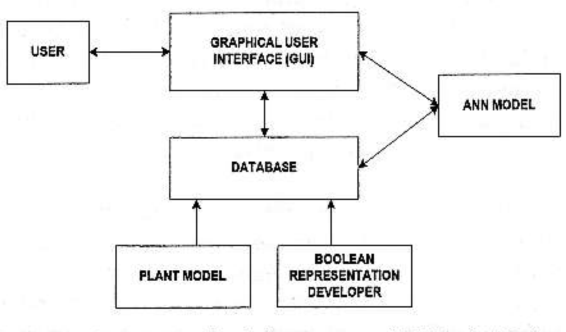

Figure 3 - Framework of the proposed PFD & AFA system.

Using normal and faulty data as the input and the respective Boolean representation as an output, the ANN model is trained. Topology analysis will be conducted to find suitable configuration that correctly recognize the fault pattern. Recall phase will also be conducted where it is subjected to a wide array of input patterns seen on training, and introduce adjustment to make it more reliable and robust. The ANN model is then validated with a novel input as to make sure the system is correct and working well.

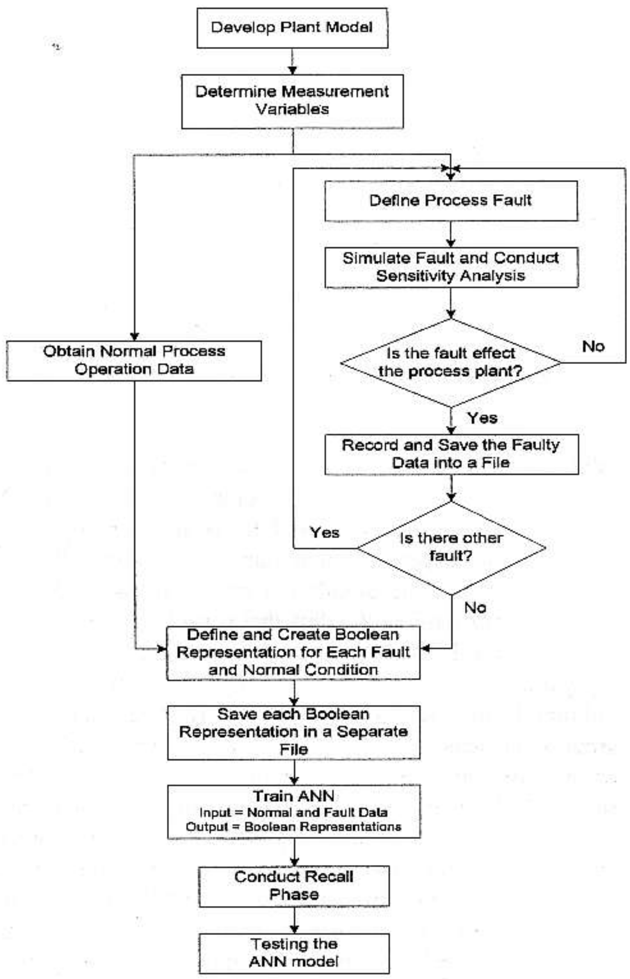

Figure 4 - Algorithm for the development of ANN model.

Case Study Description

To demonstrate the application of the proposed system in detecting fault, a pre-cut column of an actual oleochemical process plant is chosen as the study case. The oleochemical plant is a typical palm oil fractionation process where fatty acids are split from the feed materials. It consists of five distillation columns; a pre-cut column, a light-cut column, a middle-cut column, a still column and a residue still column, connected in series. Figure 5 shows simplified diagram of the process. The feed to the fractionation system are palm kernel oil and palm stearine. Because of the differences in acid distributions of the two feed, the system is operated in different modes. For this research palm kernel oil (PKO) feed system will be used as the operating mode. PKO contains fatty acids ranging from C6 to C18 constituents. In the pre-cut column C8 and C10 are recovered in the overhead. The bottom stream is fed to the light-cut column and C12 is separated from the rest of its constituents. In the middle-cut column C14 and C16 are recovered, leaving C18 to be purified by the still-cut column. The bottom stream contains some portion of C18 which is recovered in the residue-cut and recycles back to the previous column. In order to demonstrate the application of the fault detection system, the pre-cut column will be the main focused.

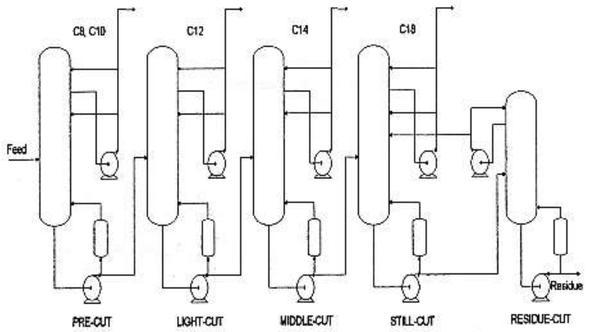

Figure 5 - Schematic Diagram of the Fractionation Process

With the advance of computer technology, commercial process simulator allows the development of plant model with ease. Plant model is used to obtain plant operation dynamics and experience that cannot be obtained through real plant data. The simulation of the plant model can provide normal and faulty data that is useful for testing and validating the PFD&D system. A process simulator called HysysPlant@ from Hyprotech is used to simulate the precut column. It provides an integrated steady-state and dynamic simulation capabilities. Due to the non-conventional nature of the palm oil distillation system, the development of the required flowsheet for plant simulation has not been straightforward. Some modifications have been made to satisfy the modeling requirement so that the output obtained from the simulation successfully mimics the actual process [6]. Figure 6 and 7 shows the main environment and column environment of the pre-cut column model plant using HYSYS, respectively.

The measurement variables to be used as input for training and testing the system are column top stage temperature, column middle stage temperature, column bottom stage temperature, column top stage pressure, reflux mass flowrate, feed mass flowrate, bottom mass flowrate and distillate mass flowrate. The dynamic measurement patterns used to train the network are obtained from plant model simulation.

Process Fault Detection

Fault Analysis

In order to identify malfunctions in the pre-cut column, various conditions are simulated and compared to the normal condition pattern. For easier comparison a graphical user interface (GUI) is developed using GUIDE toolbox within MATLAB. The GUI can help user to select different measurement variable data to be compared with the normal ones. Table 1 shows the selected conditions that effect the entire column operation and Boolean representations that represent each of the defined faults. For example, when a 5 % increase in feed flowrate (F2) was introduced and simulated, in figure 8 and 9 we can see that the column temperature and top column pressure, respectively increased to instability compared to the normal condition. Note that in the operation of the palm oil fractionation process, column temperature and top column pressure are two important variables to be monitored and controlled [7].

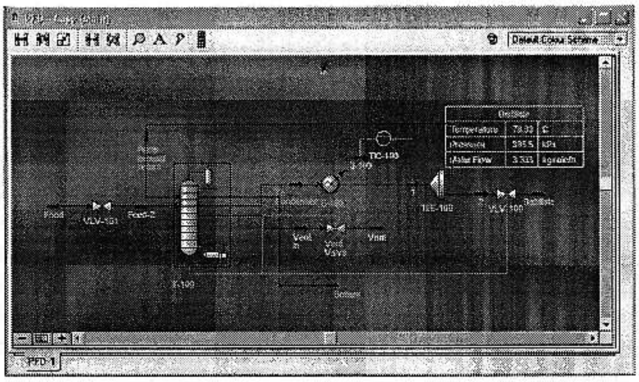

Figure 6 - Main environment.

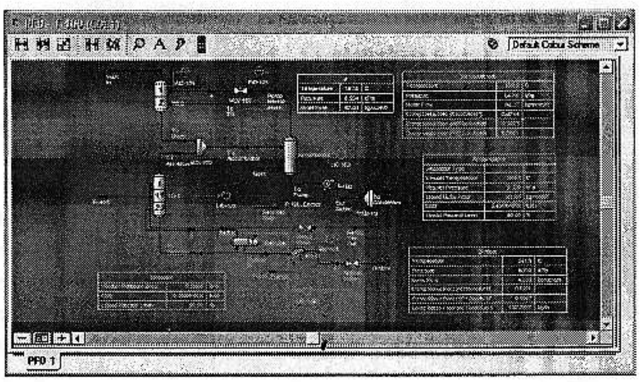

Figure 7 -Column environment.

Pattern Recognition

The development of ANN model for fault detection system was carried out using neural network toolbox available within MATLAB. A data set consist of 100 data points for each fault and its Boolean representation were used as input and output to the ANN respectively. For the input data, it was normalized between 0 and 1 before training to ensure the efficiency of the network.

To correctly recognize the fault pattern, interlayer feedforward network was selected as the connection scheme because it was the most suitable to science and engineering applications, the least complicated and very straightforward to implement. The network consists of three layers, in which *logsig* is chosen as the transfer transfer function in the hidden layer, and *purelin* transfer function was chosen for the output layer. The training was conducted using gradient descent backpropagation algorithm. To improve the generalization of the ANN, Bayesian regularization was implemented into the ANN. This algorithm works best when the network inputs and outputs are scaled in the range [1, 1].

Table 1 - Selected fault for pre-cut column.

Fault	Symbol	Boolean Representation
10% decrease in feed flowrate	F1	1 0 0 0
5% increase in feed flowrate	F2	0 1 0 0
20% decrease in feed temperature	F3	0 0 1 0
10% increase in feed temperature	F4	0 0 0 1

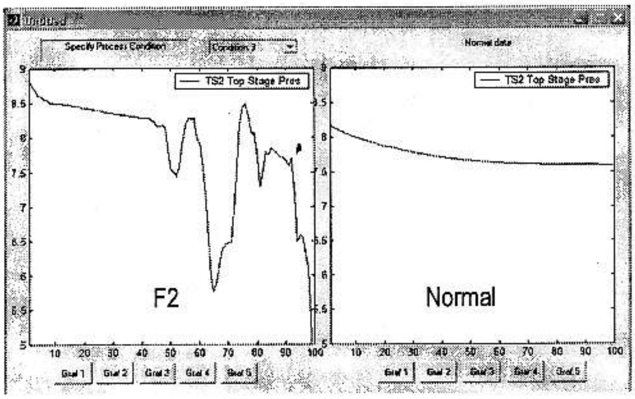

Figure 8 - Comparison between F2 and normal condition on TS2 – top stage pressure measurement data.

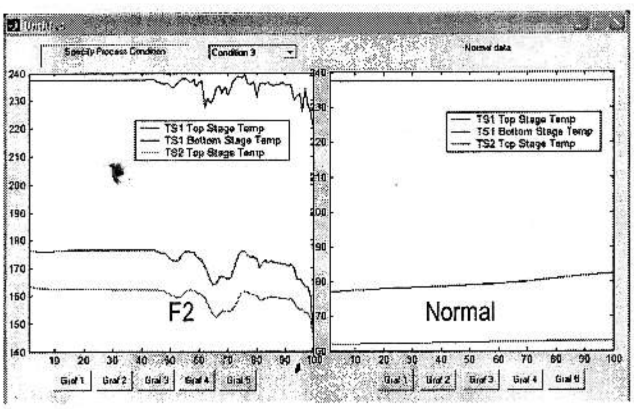

Figure 9 - Comparison between F2 and normal condition on TS1 – top stage temperature, TS1 – bottom stage temperarure and TS2 – top stage temperature measurement data.

The number of hidden layers and hidden nodes are two important factors to ensure correct classification of fault [8]. Presently, neural network with one hidden layer was used in the simulation. The number of hidden nodes required to perform accurate classification were based on the lowest value of mean square error (MSE). MSE is defined by: -

$$E = \frac{\sum_{n}\left(\sqrt{\sum_{i}(t_i - y_i)^2}\right)}{N} \; ; n = 1, 2, \ldots N \qquad (7)$$

where E_n is the MSE for data training n; N is the number of data training sets; t_i and y_i are the desired and actual output of node i, respectively. Table 2 shows values for MSE with 3, 5, 7, 11 and 15 hidden nodes.

Table 2 - MSE result with various numbers of hidden nodes

No. Hidden Nodes	MSE
3	1.925×10^{-1}
5	3.515×10^{-4}
7	4.816×10^{-8}
11	3.740×10^{-9}

From Table 2, the lowest value of MSE was achieved by a network with 11 hidden nodes. In the initial training 3 hidden nodes were used and the MSE in quite large, but when the hidden nodes were increased the MSE becomes lower. Table 2 also shows that the number of hidden nodes is important in developing an ANN that correctly classifies novel input.

After the network topology was selected, the ANN is subjected to novel input data. As mentioned earlier, each of the defined fault is represented by a specific Boolean representation. Value of '1' show that there is a fault occurred otherwise value '0' indicates there is no fault. To assist user in conducting training and validating of input, a GUI using GUIDE within MATLAB was developed. The GUI consisted of fault definition and its Boolean representation, graphs to show the classification results, training and validation buttons.

To illustrate the network capability, fault conditions F1, F2, F3 and F4 were introduced into the pre-cut column and the data obtain from the simulation was then subjected into the network to be classified. For example, when 10% decrease was introduced in feed flowrate (F1) the ANN successfully detected the fault. Similar result was obtained when 20% decrease in feed temperature (F3) was introduced into the pre-cut column. Figure 10 shows the result obtained from the GUI when F1 and F3 occur. In all cases, similar results obtained when the network was subjected to other fault pattern. The network had been successful in detecting all the failures introduced into the plant.

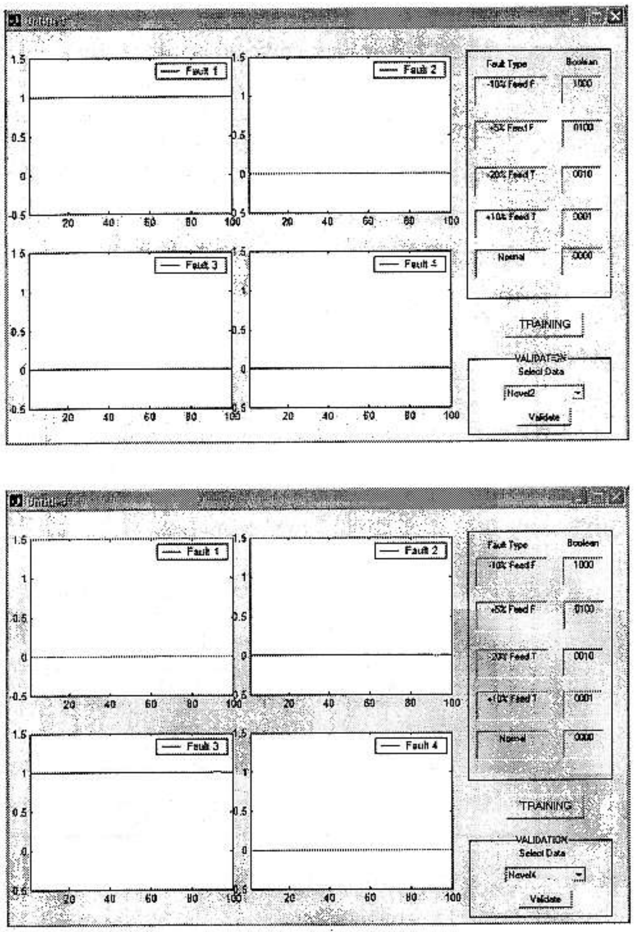

Figure 10 – Result obtained when F1 and F3 occurred.

Conclusion

A process fault detection and diagnosis on fatty acid fractionation column based on Boolean Representation using ANN has been presented. Each fault defined has been represented by a specific Boolean representation. Using the faulty data as input and the Boolean representation as output the network was trained. Network topology is important as it ensures correct classification of faults. From the results obtained, the network was successful in detecting fault occurred in the plant. ANN however, cannot offer deep investigation of faults. To extend the capability of process fault detection and diagnosis, an effort to integrate ANN with Knowledge Based Expert System (KBES) will be a subject of further work.

Acknowledgements

The authors would like to thank the Universiti Teknologi Malaysia (through the IRPA Grant Vot 74036) for sponsoring this research.

References

[1] McGraw-Hill Economics, 1985; Bureau of Labor Statistics, 1998; National Safety Council, 1999.

[2] Korbicz, J., Ucinski, D., Pieczynski, A. and Marczewska, G. (1994). "An Integrated Approach to Fault Detection and Diagnosis in Power Plant." IFAC fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland.

[3] Hoskins, J. C., Kaliyur, K. M. and Himmelblau, D. M. (1991). "Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Network." *AIChE Journal.* Vol. 37. No.1. pp. 137-141.

[4] Himmelblau, D. M. (2000). "Applications of Artificial Neural Network in Chemical Engineering." *Korean J. Chem. Eng.* 17 (4). 373-392.

[5] Venkatasubramaniam, V., Vaidyanathan, R. and Yamamoto, Y. (1990). "Process Fault Detection and Diagnosis Using Neural Networks - I. Steady-State Process." *Computers Chem. Engineering.* Vol. 14. No. 7. 699-712.

[6] Ahmad, A., Wong Teck Siang and Ling Leong Yau. (2000). "Dynamic Simulation for Palm Oil Fractionation Process."

[7] Ahmad, A. and Abd Hamid, M. K. (2002). "Detection of Sensor Failure In A Palm Oil Fractionation Plant Using Artificial Neural Network." *Proceedings of the International Conference on Artificial Intelligence in Engineering and Technology, 17-18 June 2002, Kota Kinabalu, Sabah.* pg. 739-745.

[8] Venkatasubramaniam, V., King Chan. (1989). "A Neural Network Methodology for Process Fault Diagnosis." *AiChe Journal,* Vol. 35, No. 12, pg. 1993-2002.