MODEL-BASED FAULT DETECTION USING HIERARCHICAL ARTIFICIAL NEURAL NETWORK

Arshad Ahmad and Leong Wah Heng
Department of Chemical Engineering
Universiti Teknologi Malaysia,
81310 UTM Skudai, Johor,
Malaysia.
Email: dr_aa@fkkksa.utm.my

Abstract: In this paper, a two-stage approach integrating a neural network dynamic estimator and a neural network fault classifier is proposed to overcome the problem of malfunction in sensors. The process estimator is designed to predict the dynamic behaviour of the normal or fault-free operating process even in the presence of sensor failures. The difference between this estimated "normal" values and the actual process measurements, termed the residuals are fed to the classifier for fault detection purposes. The classifier then identifies the source of faults. The scheme was implemented under dynamic operating conditions of the Tennessee Eastman challenge process and was successful in detecting various sensor faults introduced within the system.

Keywords: neural networks, fault detection, steady state model.

1.   Introduction

In recent years, the operational requirements for process industries are becoming increasingly difficult to satisfy. Stringent production specification and tight regulations imposed by local governments have led to the needs for high quality production environment. Globally competitive markets in most chemical sectors further amplify the needs for more efficient and effective process control system. As a result, new plants are now equipped with sophisticated process control facilities. Older plants are also being upgraded to include automatic control features. These facilities provide opportunities for safer and more profitable operations. However, this can only materialize if two aspects are satisfied. Firstly, suitable process control strategies must be chosen, designed and installed based on process requirements and constraints to meet the control objectives. Secondly, precise process information from the data logging facilities must be guaranteed. Whilst the design specifications are often fully satisfied during the plant installation stage, plant operation may still be subsequently unsatisfactory. For example, if incorrect information due to abnormalities in the process is fed to the control system, inappropriate control action may be taken. This will result in poor control of the plant operation leading to off-specification product generation. On occasion of serious faults, the stability limit of the plant operation might be violated leading to mandatory shutdown situation.

Whilst smart instrumentations are taking positions in today's industries, hence further improving the instrumentation technology, the ability of the control systems to detect failures accurately ahead of time is still not customary. Although some form of data reconciliation methods are available, the diagnosis of suspected faults have not been straightforwardly accomplished. As a result, alarm systems are still mostly developed based on switches. Warning and indication of failures are discrete events that are prescribed by signals form various low or high levels switches.

In practice, process faults can emerge in various different forms. Kanuri and Venkatasubramanian (1993) suggested that the following faults are significant for process plant:
1.   gross parameter changes in the actual process such as a change in the feed concentration to the reactor and fouling in a heat exchanger.
2.   structural changes such as failure of a control valve and a breakage of piping.
3.   corrupted measurement such as sensor failure, biased signals and instruments saturation.

These failures may eventually intensify to greater flaws if not detected. To alleviate such drawbacks, preventive measures should be taken using some forms of intelligent system to detect and diagnose the incoming signals. This can materialize using expert system or intelligent models within the data gathering system.

Various methods have been suggested for the detection and diagnosis of process faults. Knowledge-based Expert System received wide attention in 1980's. For examples, the works by Rich and Venkatasubramaniam (1986) and Shum *et al.* (1987) illustrate the implementation of such strategies. However, expert systems suffer from a major setback of being knowledge intensive and very time consuming to develop. Other limitations include the inability of the system to learn or dynamically improve its performance. The system is also unable to predict fault characteristics outside the domain of the prescribed expertise.

Towards the end of the 1980's, the use of Artificial Neural Network in numerous related applications such as in speech recognition, text pronunciation, virtual pattern recognition, forecasting and various types of signal and chart analysis was reported. Artificial Neural Network enjoys upper-hand advantages over expert system in terms of its development methodology, ability to generalise as well as robustness in application. Similar to linear modelling techniques, ANNs are capable of approximating the dynamic relationships between cause and effect variables. In contrast to linear techniques, ANNs are able to capture non-linear interactions among process variables. This paper elaborates the application of neural networks in fault detection and diagnosis. In the following sections, an introduction to ANN will be presented. This is will be followed by the case studies with results, discussions and conclusion drawn from the work.

2. Artificial Neural Network

Artificial Neural Network (ANN) is a computing tool that is inspired by the capabilities of human brains. An ANN is constructed of interconnected basic elements called nodes or neurons. A schematic diagram of this neuron is shown in Figure 1. Similar to their biological counterpart, these neurons are capable of processing incoming information and transferring them to other neurons.
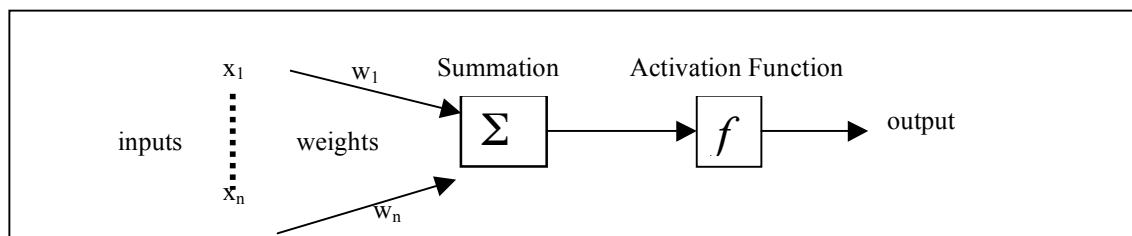

Figure 1: Schematic Diagram of an Artificial Neuron

The input signals come from either the environment or outputs of other neurons through connections as specified by the network architecture. Within each neuron, input signals are summed and transformed using an activation function before being sent to other neurons. Transformation of data via activation functions is needed to impart pattern-mapping capability to the networks. An example of such functions is the sigmoid function given by the following equation:

$$f(z) = (1 + e^{-z})^{-1} \tag{1}$$

Here, z is a weighted sum of all inputs. Associated with each connection is an adjustable value called network weights. During learning, these weights are adjusted to fulfil the training objective. Effectively, network weights serve as a measure for connection strength that controls the influence of each incoming signal on the recipient neuron.

In process engineering applications, the most commonly used network architecture is the multilayer feedforward network as displayed in Figure 2(a) below. This network is also known as multilayer perceptron. It is constructed of neurons arranged in several layers. There is an input layer to receive the incoming data to the network, and an output layer to deliver the processed data from the network. In between these two layers, there could be several layers known as the *hidden layers*. Except for those in the input layer, all neurons carry out information processing as mentioned earlier. The selection of input variables is carried out using various considerations. The aim is to include all strong relationships and neglecting some of the weaker links for the same of model simplicity. Experiences have shown that by feeding some of the delayed values of the outputs as inputs to the network, the prediction capabilities are improved significantly. The input-output relationships between the variables can be represented by equation (2) below:

$$\hat{y}_i\ (k) \quad = f\ \{u_1(k), \dots\ u_1(k\text{-}n);\ u_2(k), \dots\ u_2(k\text{-}n);\ \dots;\ y_i(k\text{-}1) \dots\ y_i(k\text{-}m)\} \tag{2}$$

where u_i is the input and $\hat{y}_i$ is the predicted output and $y_i(k\text{-}1)$ is the delayed output signals. The number of delayed signals to be used for the case of both input and output variables depends on the process. Looking from the perspective of linear modelling, decisions can be based on the model order. However, this may not be strictly followed here. The main aim is to accommodate the effect of time delays and any uncertainties associated with them.
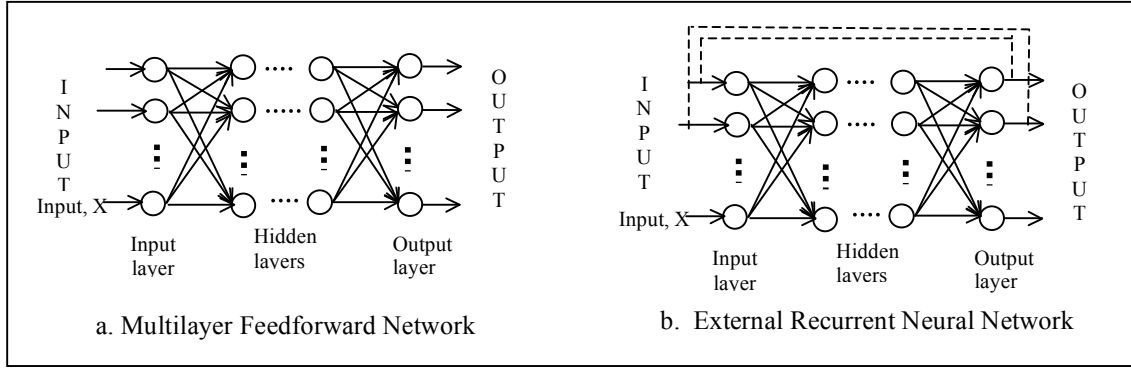

Figure 2: Examples of Neural Network Architecture

An extension of this network that is particularly useful for this work is shown in Figure 2(b). Here, instead of the delayed output signals, the delayed predicted outputs are included as input signals. The main advantage of this network is that the performance will not be influenced by the output variables. This is an important requirement of the methodology proposed in this work. Mathematically, the network can be represented by the following equation:

$$y_i(k) \quad = f\ \{u_1(k), \dots u_1(k-n); u_2(k), \dots u_2(k-n);\ \dots;\ \hat{y}_i(k-1) \dots \hat{y}_i(k-m)\} \tag{3}$$

Here, u_i is the input and $\hat{y}_i$ is the predicted output.

2.1 Network Training

An important part of the neural network model development is the training stage where the optimum connection weights are determined through some selected optimisation algorithm. During training, the optimisation algorithm (often referred to as learning rule) adjusts the network weights so that the error between the actual output and the target output is minimised. One commonly used error criterion is the mean squares error given by equation (4) below:

$$E = \frac{1}{N} \sum_{i=1}^{N} (\ t_i^{(m)} - y_i^{(m)}\ )^2 \tag{4}$$

Here, N denotes the number of training data presented to the input layer, $t_i^{(m)}$ represents the desired value of the ith output element given the mth data, while $y_i^{(m)}$ is the actual output of the same element.

As an illustration, consider a backpropagation algorithm, which is a simple gradient-based technique that has been widely used. The weight update equation for this class of algorithm can be represented by equation (5) below.

$$W_{ji}^{(m)} \quad = \ W_{ji}^{(m\text{-}1)} + \ \Delta W_{ji}^{(m)} \tag{5}$$

$W_{ji}^{(m)}$ denotes the weight of the connection between the jth element of the upper layer and the ith element of the lower layer in the mth learning iteration. For example, in the case of back propagation algorithm, the required weight change $\Delta W_{ji}^{(m)}$ is calculated as follows:

$$\Delta W_{ji}^{(m)} \quad = \ \eta \ . \ \delta_j^{(m)} \ . \ O_i^{(m)} \ + \quad \alpha \ \Delta W_{ji}^{(m\text{-}1)} \tag{6}$$

Here, η and α denote the learning rate and coefficient of the momentum term, $O_i^{(m)}$ is the output value of the *i*th element in the previous layer and $\delta_j^{(m)}$ is the gradient descent term. Other gradient-based methods also implement similar strategies, the main difference being the computation of the required weight change (i.e., equation (6)). In order to overcome some of the limitation encountered when using any first order gradient-based optimisation technique, higher order methods are now more commonly used.

2.2 Selection of Network Topology

Another important stage in neural network model development is the selection of network topology, i.e., the number of neurons in each of the network layer and its connection. Once the type of the network has been chosen, e.g., multilayer feedforward network, the orientation of network connection is defined. Similarly, the number of neurons in the output and input layers are also defined by the problem and hence, the emphasis on the topology selection is to determine the number of neurons in the hidden layer. The aim is to develop models that are accurate and robust. Neural network is known to be able to map any continuous function to any arbitrary accuracy (Cybenko, 1989). This implies that the network can learn the relationship between any set of inputs and outputs so that when given the inputs, the outputs can be reproduced. To obtain sufficiently good approximation qualities, a network with "sufficient" neurons must be trained. In doing so, that the weights associated with all the connections within the network are optimised to achieve the desired input/output mapping. In developing the model, what is important is the ability of the model to predict the behaviour on "unseen" process. Therefore, the decision on the "optimal" model structure should not be made based on the performance of the model to reproduce the training set only because the resulting model may not fulfil the robustness requirement. Although there are many methods or criteria that are available to determine the best fit of the model parameters, in this work cross validation approach is adopted.

The concept of topology selection using cross-validation is that after estimation using a given sample of data, the quality of the mapping is evaluated using a different set of data. The best mapping is defined as the one that minimizes the prediction error on a data set for which it was not trained. This approach of topology selection involves iterative efforts. Beginning with some small number of hidden neurons, the search continues until the desired performance is achieved.

3.0   Process Description

The application of neural network in fault detection and diagnosis is implemented to the Tennessee Eastman (TE) plant (Downs and Vogel, 1993). In this plant, two products (G, H) are produced from four reactants (A, C, D, E). Also present in the process is an inert B and a by-product F. Figure 3 shows the schematic diagram of the process. The plant is consisting of five major unit operations: a feed mixing unit, a reactor, a vapour-liquid separator, a product stripper and a recycle compressor.
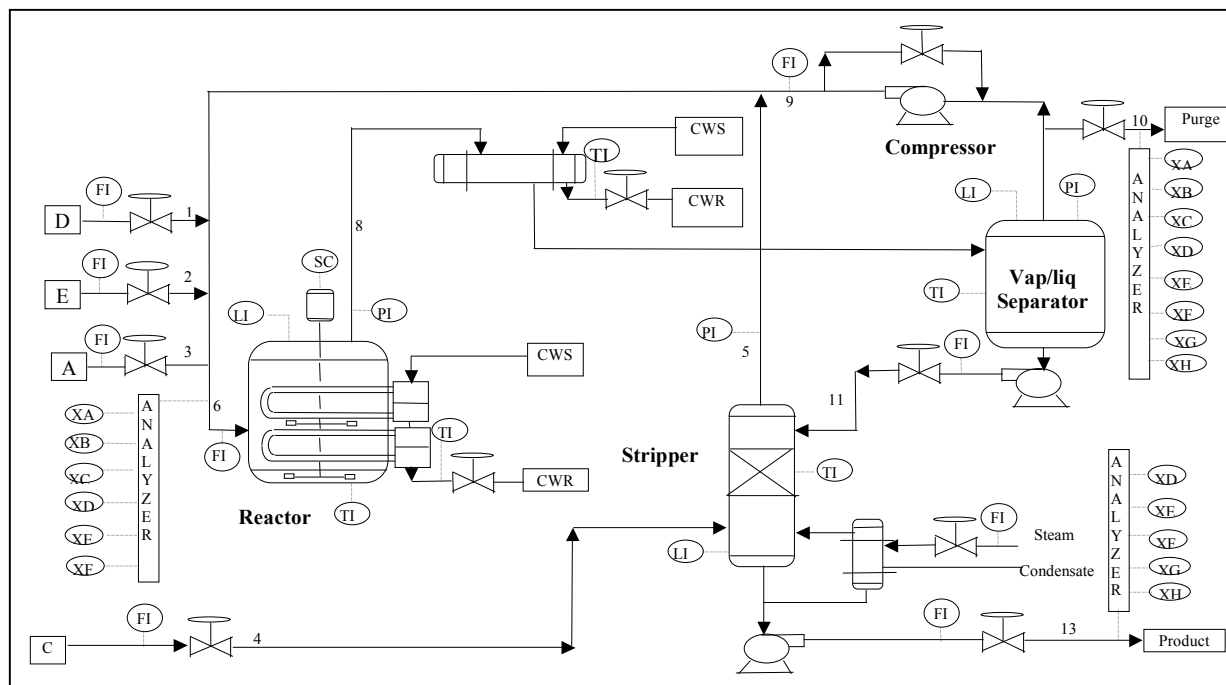

Figure 3 : Schematic Diagram of Tennessee Eastman Plant

Gaseous reactants are fed to the reactor in which they react to form liquid products. The product leaves the reactor as vapours along with unreacted feeds through a stream that passes through a cooler and then to a vapour-liquid separator. Noncondensible components are recycled through a centrifugal compressor to the reactor feed. Condensed components are fed to a product-stripping column to remove remaining reactants by stripping with feed stream number 4. Products G and H leave the stripper base to be separated in a downstream refining section. The inert and by-product are primarily purged from the system as a vapour from the vapour-liquid separator.

This investigation focused on the malfunctions of the process caused by the failure of the level, temperature and pressure sensors in the reactor. Faulty conditions are simulated using the Tennessee Eastman Plant. Sensor failures are created causing the normal process operation to shift to a faulty operation mode. Effects of these faults are expressed by the composition of components A to H in the purge stream and the composition of components D,E,F,G,H in the product stream leaving the stripper. During implementation, sensor failures are categorised into high and low readings.

4.0 Neural Network Fault Detection Scheme

The proposed fault detection scheme is hierarchical in structure. The schematic diagram of the strategy is shown in Figure 4 below. In short, there are two types of model developed using neural network. The first is the predictor that will always predict the "normal" or fault-free process behaviour. Next, there is the fault classifier that will identify the sources of fault that takes place.
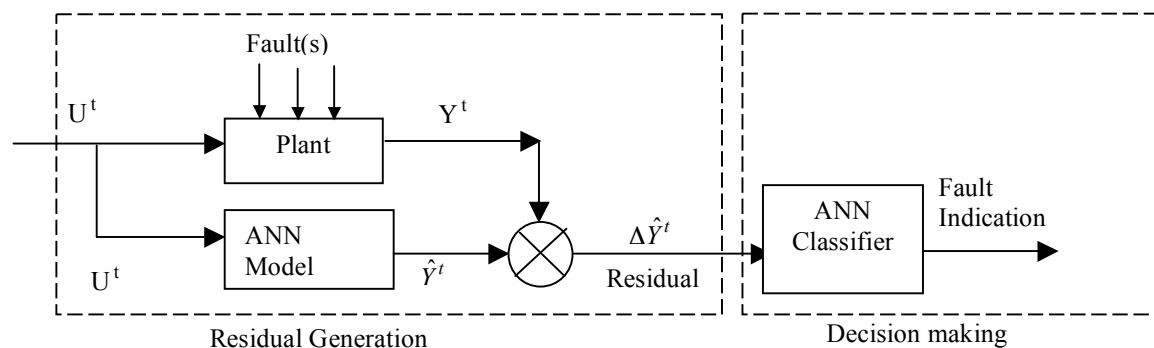

Figure 4: The Model-Based Fault Detection

The hierarchical approach is advantageous because it alleviates the chances of misidentification of normal operation trend that is due to the manipulation of the feeds condition. In practice, there are always possibilities that the manipulation of feeds will produce process conditions that coincidentally match the fault pattern and the classifier will tend to misinterpret the situation. The use of residuals provides some protection to the system.

4.1 Process Predictor

The model structure for Process Predictor can be either consisted of a set of Multi-Input Single-Output (MISO) network or single Multi-Input Multi-Output (MIMO) network. The selection depends on how efficient are the structures in predicting the process. For MISO structure, three different models are required to predict the three outputs of the process. On the contrary, only one MIMO model is required to accomplish the same task. The key deciding factor between these two strategies are the effort as well as accuracy of models produced. The MISO models can be represented by equation (5) below:

$$\hat{y} = f(\hat{y}_i(t-1), u_1(t-1), u_1(t-2), u_2(t-1), u_2(t-2)\,)\,) \tag{7}$$

Here, u_1 and u_2 are the D feed and A+C feed flowrates respectively and y_i's are the process outputs, i.e., reactor pressure, product composition or separator temperature. Since the previous output data cannot be used as one of the input as it will influence the prediction significantly, only the delayed signal of the predicted output can be added as the network input. Such networks fall into the category of recurrent networks and as such, standard training method cannot be directly used. In this work, a two-stage training approach was employed. First, the network was trained

using Levenberg-Marquadt algorithm assuming that the delayed signal of the actual output was being used. This was followed by a retraining using dynamic backpropagation, which is one of the methods introduced for training of recurrent network. Based on the results obtained, the minimum cross-validation error is conceded by MISO networks with 5,25,15 hidden nodes for reactor pressure, G% in final product and separator temperature respectively.

The MIMO network structure compresses all the input and output variables in one network model. Two delayed input data for each input variable and one delayed network output for recurrent input made up the input layer of the network. The model representation of MIMO network can described as:

$$\hat{y}_i(t) = f_i\big(\hat{y}_1(t-1), \hat{y}_2(t-1), \hat{y}_3(t-1), u_1(t-1), u_1(t-2), u_2(t-1), u_2(t-2)\big) \tag{8}$$

The results revealed that the optimum network structure was established with 13 hidden neurons. Figure 5 displays the predicting capabilities of the network developed. In this study, MIMO structure was found better, and thus employed in the fault detection scheme.
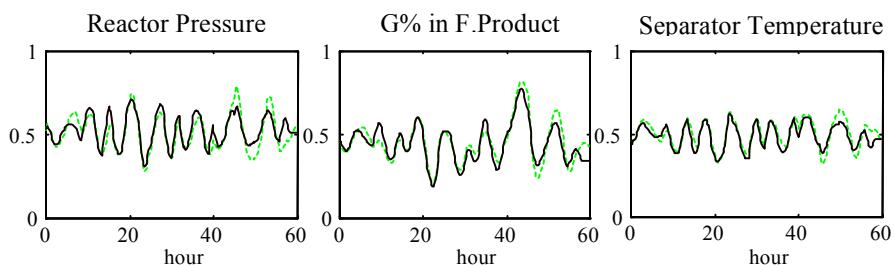

Figure 5: Performance of MIMO network (topology 7/13/3) predicting unseen

4.2 Fault Classifier

The fault classifier was constructed using MLFF network with single hidden layer. As mentioned earlier, the MLFF network can be either composed of a set of MISO networks or a single MIMO networks. Again, in this study both network structures were examined. Network training was implemented using Levenberg-Marquardt learning algorithm. Training become sufficiently fast with batch training compared to the pattern training that was employed in Dynamic Backpropagation method for the process predictor. The network was also cross-validated at every batch training and thus the cross-validation errors of the network were monitored throughout the training. Network weights and biases were selected based on the minimum cross-validation error achieved in the training.

This work focused on the detection of corrupted sensor measurement. Sensor faults were simulated as if there were sudden failures of the measurement systems resulting in measurements bias. The extent of the sensor bias were used as indicators for severity of the sensor fault. Since in the operation of the TE process, reactor temperature sensor and reactor pressure are the two important variables to be controlled, these variables were considered as the focus of this work. Both positive and negative bias measurements of these sensors were simulated for fault detection. Table 1 shows the list of process faults involved.

Table 1 List of sensor faults

1	Reactor Temperature Sensor + ve bias	F1
2	Reactor Temperature Sensor - ve bias	F2
3	Reactor Pressure Sensor +ve bias	F3
4	Reactor Pressure Sensor –ve bias	F4

All these faults influence not only the performance of the control loops within the plant, but also the stability of the process. For example, when F3 with 4 % bias was introduced, the reactor pressure increased exponentially to instability. Some of these impacts are shown in Figure 6 below.

The output data for sensor faults were designed to spread linearly between 0.0 and 1.0. Zero and one were used to represent the process during normal condition and violation of process limit respectively. Similar to the case of the predictor, both MIMO and MISO models were considered. The MIMO network structure employed the reactor pressure, composition of product G and the separator temperature as the input. The outputs were the various faults F1, F2, F3 and F4. Again, a feedforward network with one hidden layer was used. Training was accomplished

using Levenberg-Marquadt algorithm and a network with 12 nodes in the hidden layer was considered optimum. For the MISO structure, four networks, each responsible for a fault class were developed. The results of network training revealed that a network with 9,17, 18,11 hidden nodes for classifier F1, F2, F3 and F4 respectively are optimum. In this case, MISO networks showed better generalisation ability compared to MIMO network and thus adopted.

(a) Reactor Pressure at F1 (2.0%)

(c) Reactor Pressure for F3 (4.0%)

(b) Reactor Pressure for F2 (5.0%)

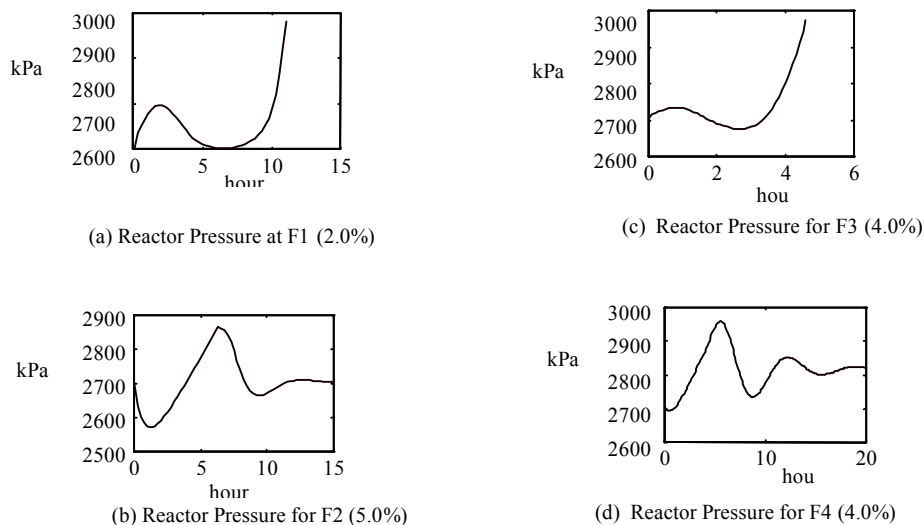(d) Reactor Pressure for F4 (4.0%)

Figure 6: Effect of sensor faults to the dynamics of the reactor pressure

In order to examine the capability of the proposed scheme, faulty conditions were simulated under dynamic process condition. The results obtained revealed the success of the classifiers in detecting the faults introduced to the system. For example, Figure 7 illustrates the detection of Fault 1. When a 1% increase in flow rate of D feed stream and a 5% bias introduced to reactor temperature sensor the classifier F1 successfully detected the fault. This is clearly displayed in Figure 7 where the network output reached index of 0.8, as designed to indicate a process fault.
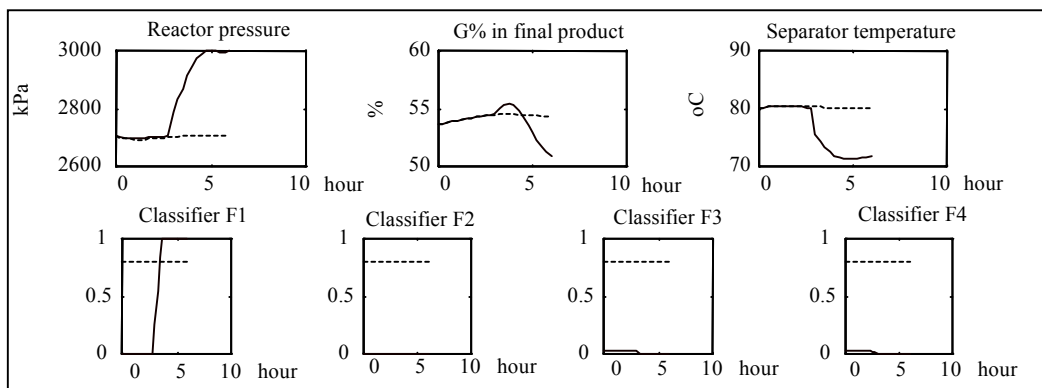

Figure 7: Detection of fault with 1 % increase in Flowrate of D Feed

Similarly, when the flowrate of A+C feed stream was increased by 1%, the reactor encountered negative bias of 7%. The outputs of classifier F2 (-ve bias in temperature) passed the index 0.8 to acknowledge the faulty condition as shown in Figure 8. Again, the process fault was successfully detected. Similar observations were established when other fault patterns were introduced. In all cases, the proposed scheme had been successful in detecting the failures. However, due to lack of space, the results are not shown here.
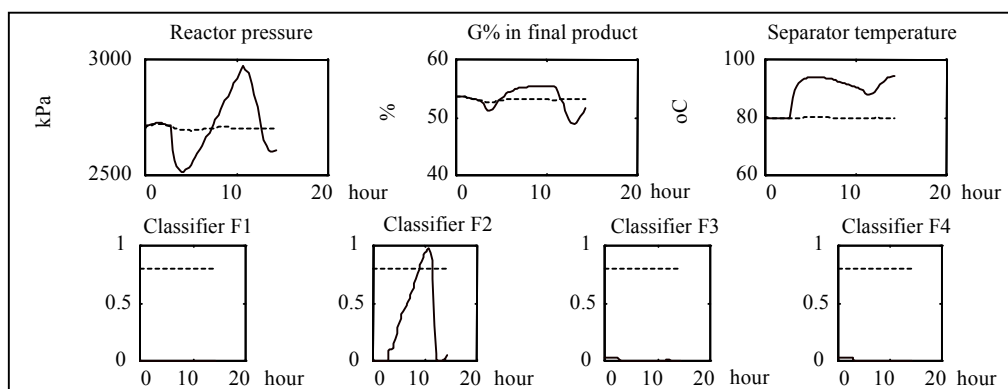
Figure 8: Detection of fault pattern with 1 % increase in Flowrate of A+C Feed

5. Conclusion

This paper has revealed the capabilities of neural networks in process fault detection. Neural network models have effectively been applied as both the process predictor and fault classifier. The proposed hierarchical structure has also been proven practicable. The dynamic detection scheme presented here is useful in fault monitoring because rather than having a discrete detection, continuous detection allows operators to anticipate potential process failures. This is better than having just a fix-value alarm signals as widely practiced in the industry.

REFERENCES

Downs J. J. , E. F. Vogel (1993). A Plant-Wide Industrial Process Control Problem. *Computer Chem. Eng.*, **17**, 245-255

Fan, J.Y., M. Nikolaou, and R.E. White (1993). An Approach to Fault Diagnosis of Chemical Processes via Neural Networks. *AIChe J.*, **39**, No.1, 82-39.

Hornik, K., M. Stinchcombe, and W. White (1989). Multilayer Feedforward Neural Networks are Universal Approximators. *Neural Networks*, **2**, No.5, 359.

Hunt, K.J. D. Sbarbaro, R. Zbikowski and P.J. Gawthrop (1992). Neural Networks for Control System: A Survey. *Automatica*, **28**, No. 6, 1083-1112.

Kanuri S.N. , Venkatasubramanian V. (1993). *Neural Network Decomposition Strategies For Large Scale Fault Diagnosis.* Technical Report, Computer Integ. Process Operation Center, Purdue University.

Rich S.H. and Venkatasubramanian V. (1986). Model-based Reasoning In Diagnostic Expert Systems For Chemical Process Plants. *Computer Chemical Engineering*, vol.11, no.2 , 111 – 122

Shum S.K. , Davis J.F. , Punch W.F. and Chandrasekaran B. (1987).   An Expert System To Mulfunction Diagnosis In Chemical Plant. *Computer Chemical Engineering*, vol.12 , no.1 , 27 – 36

Venkatasubramaniam V. (1990). Process Fault Detection And Diagnosis Using Neural Networks - I. Steady-State Processes. *Computers chem. Engineering.* **14,** 699-712