I&ECresearch
Industrial & Engineering Chemistry Research

A self-learning fault diagnosis system based on reinforcement learning

Yih Yuan Hsu, and Cheng Ching Yu

Ind. Eng. Chem. Res., **1992**, 31 (8), 1937-1946 • DOI: 10.1021/ie00008a015

Downloaded from http://pubs.acs.org on November 28, 2008

Industrial & Engineering Chemistry Research is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

A Self-Learning Fault Diagnosis System Based on Reinforcement Learning

Yih-Yuan Hsu and Cheng-Ching Yu*

Department of Chemical Engineering, National Taiwan Institute of Technology, Taipei, Taiwan 106, R.O.C.

In recent years, the evolution of qualitative physics has lead to the rapid development of deep-model-based diagnostic systems. Yu and Lee integrated quantitative process knowledge into a deep-model-based diagnostic system. In the qualitative/quantitative knowledge-based systems, the qualitative model and approximated numerical values are needed to construct a diagnostic system. This results in a bottleneck in the knowledge-acquisition step. On the other hand, another branch in artificial intelligence, artificial neural network (ANN), has the advantage of self-learning. This work utilizes the self-learning feature of the ANN such that the semiquantitative knowledge can be integrated into the qualitative/quantitative model in the learning steps. A chemical reactor example is used to illustrate the advantages of the proposed diagnostic system. Simulation results show that the proposed diagnostic system not only has the self-learning ability of ANN but also is transparent to the users. Moreover, it does not produce erroneous solutions when compared with the backpropagation ANN and it also gives less spurious solutions when compared with qualitative model-based systems.

1. Introduction

In recent years, the complexity of modern chemical plants and the availability of inexpensive computer hardware prompted us to develop automated fault diagnosis instead of conventional diagnosis by the operator (Himmelblau, 1978; Isermann, 1984; Frank, 1990). Generally, depending on the rigorousness of the process knowledge employed, techniques for automated fault diagnosis can be classified into qualitative, qualitative/quantitative, and quantitative approaches. The qualitative approach only considers the signs of coefficients in all governing equations of process variables. The signed directed graph (SDG) is a typical example. Upon diagnosis, the consistency of the branches of a given fault origin is checked to validate (or invalidate) this hypothesis and all possible fault origins are screened. In many cases, it simply gives multiple interpretations for a single event (Kramer and Palowitch, 1987; Chang and Yu, 1990). This is an inherent limitation of the qualitative model-based systems. Since only qualitative knowledge is employed, the diagnostic resolution can only be improved to a certain degree.

The quantitative model-based diagnostic systems, on the other hand, utilize the process model and on-line measurements to back-calculate crucial process variables. It finds the fault origins according to the perturbations in the calculated variables (Willsky, 1976; Isermann, 1984; Petti et al., 1990). Generally, this approach is too time-consuming and requires a significant amount of modeling effort. Originated from the artificial neural network (ANN), the backpropagation neural network is often employed in fault diagnosis (Watanabe et al., 1989). Generally, this type of approach can also be classified as a quantitative model-based diagnostic system. It utilizes a set of process data, such as the values of steady-state process variables for the nominal operating condition and these for the identified faulty conditions, to train the network (Watanabe et al., 1989; Venkatasubramanian et al., 1990; Ungar et al., 1990). Following the training, the model is established and ready for fault diagnosis. Despite its black-box nature, the backpropagation neural network can accurately pin down the fault origin in most cases. Unfortunately, the parameters (such as input variables, number of processing elements, and learning constants) must be determined by trial and error. If these parameters are not chosen adequately, the convergence of the network can be difficult and erroneous interpretations may result.

Another approach is the qualitative/quantitative model-based diagnostic system. Yu and Lee (1991) integrated semiquantitative knowledge (e.g., steady-state gains) into a deep model-based diagnostic system to improve diagnostic resolution. One advantage of this approach is that the semiquantitative knowledge is added to a qualitative model (structure). The approach of Yu and Lee is similar, in concept, to the approaches of data interpretation (Cheung and Stephanopoulos, 1990a,b; Prasad and Davis, 1991; Rengaswamy and Venkatasubramanian, 1992) which received quite a bit attention recently. The data interpretation approaches devise a mechanism to map from quantitative data to qualitative interpretations which can then be used by some appropriate qualitative (or semiquantitative) models. However, these two approaches differ significantly in defining the boundary between the model and input to the model. In data interpretation, quantitative data are transformed to qualitative (or semiquantitative) interpretations for corresponding models. As for the approach of Yu and Lee (1991), the quantitative data are plugged directly into the semiquantitative model to check the consistency. Regardless of the approaches employed, the semiquantitative knowledge has to be modified as the operating condition changes. This can lead to a knowledge-acquisition bottleneck in any realistic application. Therefore, an efficient method to acquire semiquantitative knowledge is necessary for fault diagnosis in the chemical process industries.

An ideal diagnostic system should have at least the following properties:

Soundness (Kuipers, 1988): Regardless of the number of the spurious solutions, it cannot have any erroneous solution (i.e., the true fault origin is not included in the solution set) at different operating conditions.

Transparency: The model should be easy to understand and the knowledge base, e.g., semiquantitative information, should be easy to maintain.

Self-learning: The system should be able to learn (or modify) from the process data to cope with frequently changed operating conditions.

Under any circumstance, the first property is the min-

* To whom correspondence should be addressed.

0888-5885/92/2631-1937$03.00/0

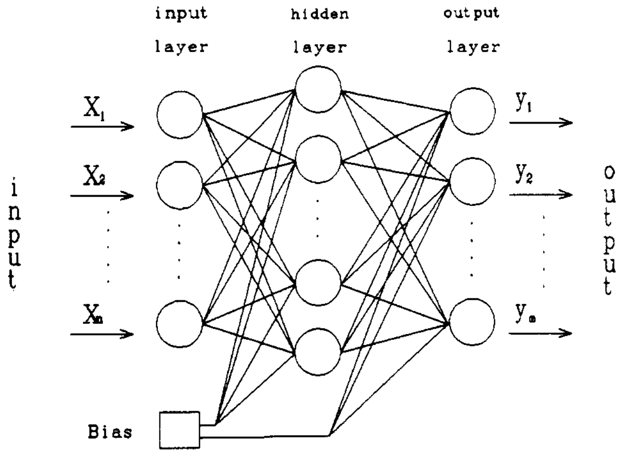

Figure 1. Multilayered feedforward neural network.

imal requirement of a diagnostic system for any practical application.

The purpose of this work is to provide a self-learning feature to the qualitative/quantitative model-based diagnostic system. The self-learning procedure is based on reinforcement learning of the neural network (Barto et al., 1983). Comparisons will be made between the qualitative/quantitative model-based system from reinforcement learning and the well-celebrated quantitative model-based system, artificial neural network using backpropagation learning. A CSTR example will be used to illustrate the model building, self-learning, and performance of these two systems. This paper is organized as follows. ANN with backpropagation learning and reinforcement learning is introduced in section 2. Section 3 describes how to add the self-learning feature to the qualitative/quantitative model. Model-based diagnostic systems are given in section 4. A CSTR example is used to illustrate the characteristics of these two systems in section 5, followed by the Conclusion in section 6.

2. Artificial Neural Network

An artificial neural network (ANN) is trained to produce a desired output by adjusting the weights on the connections between nodes according to some prespecified criteria. Generally, three types of learning procedures exist: (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning. The relevant two, the supervised and reinforcement learning, are described here.

2.1. Supervised Learning. ANN with backpropagation learning is a typical example of supervised learning. In the supervised learning, an external target output vector is required for each input vector. A common procedure is to adjust the values of the weights such that the sum of the squares of the deviations between the target value and each actual output is minimized (Rumelhart and McClelland, 1986; Lippmann, 1987). Figure 1 shows a multilayer feedforward neural network architecture. The circles (nodes) represent the processing elements in different layers: input, hidden, and output layers. Each input unit is connected to each hidden unit and each hidden unit is also connected to each output unit. Each hidden and output unit is also connected to a bias. The bias with the value of 1 is used in this work. Each connection has a weight associated with it. For the input layer, an input value is straightly forwarded to the next layer. The hidden and output units carry out two calculations. Firstly, a weighted sum of the inputs is taken (e.g., a_{ij}), and then the output is calculated using a nondecreasing and differentiable transfer function $f(a_j)$ (Figure 2). Usually the transfer function $f(a_j)$ is a sigmoid logistic function as shown in Figure 3.

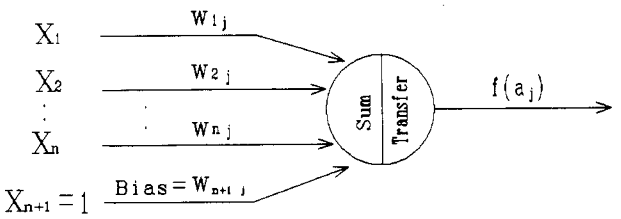

Figure 2. Processing element (neuron).

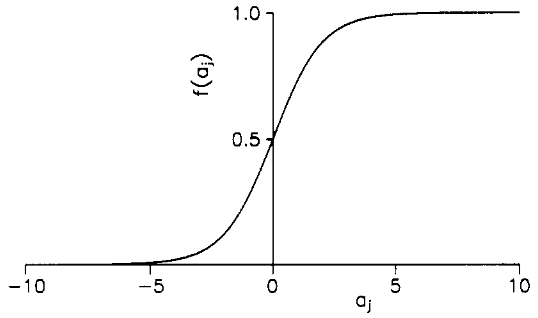

Figure 3. Processing element output transfer function.

$$f(a_j) = 1/(1 + e^{-a_j}) \qquad (1)$$

Typically, the learning rule for adjusting the weights is the generalized delta rule (GDR) (Rumelhart and McClelland, 1986). It uses the gradient-descent method to minimize the objective function E (or the mean square error):

$$E = \frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{N}(d_i^m - y_i^m)^2 \qquad (2)$$

where M is the number of training patterns presented to the input layer and N is the number of units in the output layer, d_i^m represents the target output value of the ith output element given the mth pattern, while y_i^m is the actual output of the ith unit.

For a given pattern, the weight is adjusted according to GDR as the following:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \qquad (3)$$

where $w_{ij}(t+1)$ denotes the weight of the connection between the ith element of the lower layer and the jth element of the upper layer in the $(t + 1)$th learning iteration. The weight change $\Delta w_{ij}(t)$ in eq 3 is calculated according to

$$\Delta w_{ij}(t) = \eta\delta_j x_i + \beta\Delta w_{ij}(t-1) \qquad (4)$$

where η and β are the learning rate and momentum constant; x_i is the output value of the ith element in the lower layer. The momentum term β prevents divergent oscillation and makes the convergence more rapidly. The error term of the jth element δ_j in eq 4 is determined as follows. If the subscript j denotes the output layer, then

$$\delta_j = (d_j - y_j)f_j'\sum_i(w_{ij}x_i + \theta_j) \qquad (5)$$

and if j denotes the hidden layer, we have

$$\delta_j = f_j'\sum_i(w_{ij}x_i + \theta_j)\sum_k\delta_k w_{jk} \qquad (6)$$

where f_j' is the derivative of the jth transfer function as described previously, θ_j is the bias of the connection in the jth element, and k is the upper layer element of the jth element.

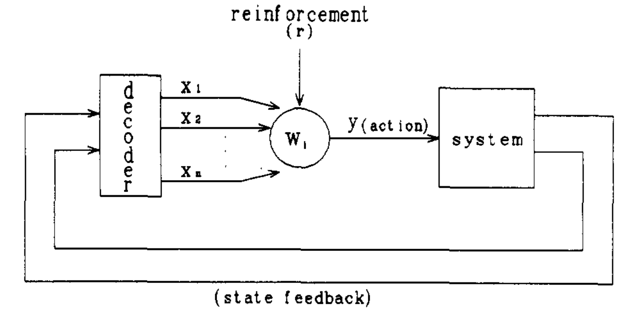

Figure 4. "Boxes" system.

According to the generalized delta rule (GDR), the network has the following training steps:

1. Initialize the weights randomly and specify the bias.
2. Specify the input patterns and the target output patterns.
3. Calculate the actual output pattern.
4. Adjust the weights by GDR.
5. Check the convergence criterion; if it is satisfied then go to step 6, otherwise go back to step 3.
6. Stop.

2.2. Reinforcement Learning. Another learning algorithm of ANN is the reinforcement learning (Barto et al., 1983). Typical applications of reinforcement learning are in control problems (e.g., fuzzy control of a cart–pole system; Lee, 1991). It uses a neuronlike element to solve the specified problem. Usually this element is called associative search element (ASE). One important difference between the supervised learning and the reinforcement learning is that the supervised learning must have a target value to correct (e.g., minimizing) the error between the actual output value and the target value. If the environment is unable to provide the appropriate response for a desired performance, then the ASE must discover what response can lead to an improvement in the performance. It employs a trial and error procedure to search for appropriate action and finds an indication of the performance. The appropriateness of such action can be judged from the performance measure. If the value of output units action is bad (does not lead to improvement), then it adjusts the weight by some strategy to keep getting good results. This is similar to a child learning to taste candy by trial and error until he finds what he likes.

In the "Boxes" system (Barto et al., 1983), the ASE is employed in a self-learning controller to control a cart–pole system. The element has a reinforcement input pathway, n pathways for nonreinforcement input signals, and a single output pathway as shown in Figure 4. The decoder divides the continuous output signal into discrete states. The element's output $y(t)$ is determined from the input vector $\mathbf{X}(t) = [x_1(t), x_2(t), ..., x_n(t)]$ as follows:

$$y(t) = f[\sum_{i=1} w_i(t)x_i(t) + \text{noise}] \qquad (7)$$

where f is the following threshold function (a step function):

$$f(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \qquad (8)$$

The weights w_i's are changed according to

$$w_i(t+1) = w_i(t) + \alpha r(t)\, e_i(t) \qquad (9)$$

$$e_i(t+1) = \delta e_i(t) + (1 - \delta)y(t)\, x_i(t) \qquad (10)$$

where α = learning rate, δ = trace decay ratio, r(t) = reinforcement signal at time t, $e_i(t)$ = eligibility at time t of input pathway i, and $x_i(t)$ = input vector at time t. Specifically, the Boxes system divides the system inputs into many substrates (input pathways) by the decoder. The system performance is judged according to the inputs. If the performance is bad, then the system gives a reinforcement signal r. Whenever certain conditions hold for the ith input x_i, then this pathway becomes eligible to have its weight modified. In the Boxes system, the input x_i triggers the eligibility trace whenever the box i is entered. According to r and the eligibility, a better performance is sought by changing the output (action) via the adjustment of the corresponding weight.

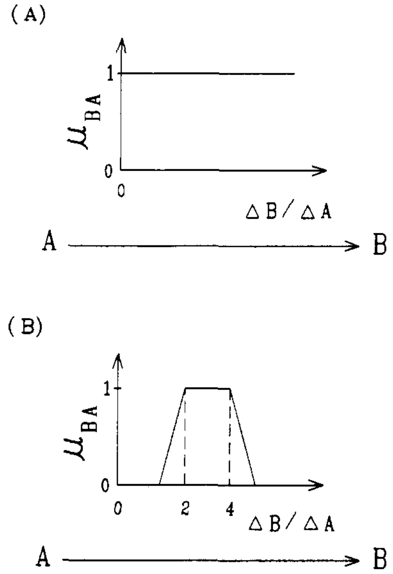

Figure 5. Membership functions for (A) qualitative model and (B) qualitative/quantitative model.

3. A Self-Learning Qualitative/Quantitative Model

3.1. Qualitative/Quantitative Model. In qualitative reasoning, the diagnostic resolution is limited by the strictly qualitative knowledge. Yu and Lee (1991) integrated the semiquantitative knowledge into a qualitative model using fuzzy set theory. The shape of the membership function represents the semiquantitative information between process variables. Consider a simple qualitative model: the signed directed graph (SDG), e.g., $A \xrightarrow{+} B$. The binary relation between A and B can be described by the ratio $\Delta B/\Delta A$ taking the value from 0^+ to infinity. In terms of the qualitative/quantitative model, the membership function $\mu_{BA}(\Delta B/\Delta A)$ takes the value of 1 for all positive $\Delta B/\Delta A$ as shown in Figure 5A. If some semiquantitative information is known, e.g., the steady-state gain between A and B falls between 2 and 4, we can modify the membership function accordingly (Figure 5B). However, the construction of the semiquantitative knowledge requires a great deal of engineering effort, even when all process data are available. Furthermore, the semiquantitative knowledge needs to be modified as we change the operating conditions. One important advantage of the qualitative/quantitative model is that the qualitative part of the model (the structure) remains the same under almost all possible operating conditions. Therefore, when the operating condition changes, all one has to do is to modify the semiquantitative part of the process knowledge.

3.2. Self-Learning Feature via ASE. Reinforcement learning is employed to acquire the semiquantitative knowledge automatically. The basic idea is shown is Figure 6. For a given fault origin, we can find measurement patterns from process simulation or past events. The measurement patterns are fed into the qualitative/quan-

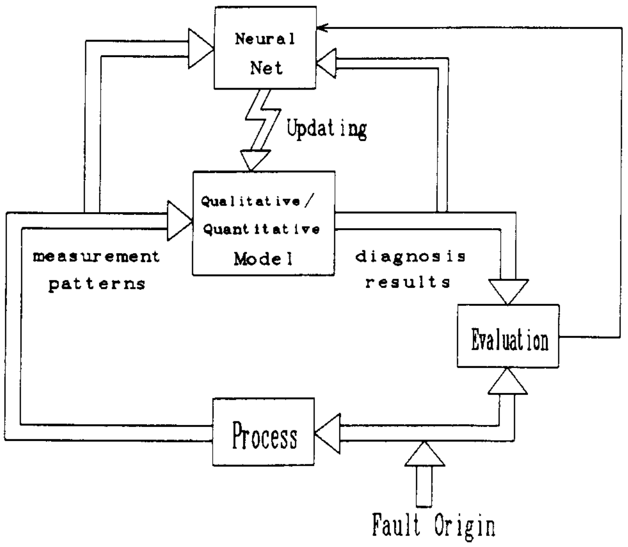

Figure 6. Schematic representation of a self-learning process for the qualitative/quantitative model-based diagnosis system.

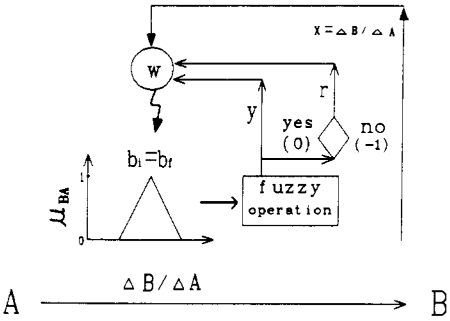

Figure 7. Schematic representation of reinforcement learning via information on qualitative/quantitative model.

titative model. If the fault is not correctly identified, the reinforcement learning is activated and the measurements and the diagnostic results are fed to the corresponding ASE. Subsequently, the shape and location of the membership function is changed until a satisfactory diagnostic result is found (Figure 6). That is, the model learns from the measurement patterns repeatedly until good performance (correct diagnosis) is achieved. In the meantime, the membership function in the model (the semiquantitative knowledge) is adjusted to ensure good performance. Let us take a single branch between nodes A and B and its corresponding ASE as an example (Figure 7). Initially, the membership function is located according to the measurement $\Delta B/\Delta A$. This value corresponds to the full membership, i.e., $\mu_{BA}(\Delta B/\Delta A) = 1$, and the membership decreases linearly to zero for ±20% deviations in $\Delta B/\Delta A$. If another set of measurement pattern is available and the result of the fuzzy operation is not satisfactory, e.g., $\mu_{BA}(\Delta B/\Delta A) \neq 1$, then the system responds with a reinforcement signal $r = 1$. This indicates the location and shape of the membership function is incorrect (Figure 7). Therefore, the ASE *reshapes* the membership function according to $\Delta B/\Delta A$ and $\mu_{BA}(\Delta B/\Delta A)$ until a good performance $\mu_{BA}(\Delta B/\Delta A) = 1$ is achieved. The membership function is relocated according to the weight change Δw ($\Delta w = w(t+1) - w(t)$). In this work, the weight w is changed according to

$$w(t+1) = w(t) + \alpha r(t)\, e(t) \tag{11}$$

$$e(t+1) = \delta e(t) + (1 - \delta)(1 - y(t))x(t) \tag{12}$$

where α = learning rate, δ = trace decay ratio, $r(t)$ = reinforcement signal at time t, $e(t)$ = eligibility at time t, and $y(t)$ = the result of fuzzy operation at time t. The ASE adjusts the location and shape of the membership function in the following way:

$$f(x) = \begin{cases} |\Delta w| & \text{if } x > b_f \\ 0 & \text{if } b_i \le x \le b_f \\ -|\Delta w| & \text{if } b_i > x \end{cases} \tag{13}$$

where b_i and b_f are the initial and final values of the process measurements satisfying $y = 1$. That is, b_i and b_f correspond to the upper left and right corners of the trapezoid shaped membership function.

As shown in Figure 7, the following information (1) process measurement x_i (or $\Delta B/\Delta A$), (2) the result of fuzzy operation y (or $\mu_{BA}(\Delta B/\Delta A)$), and (3) reinforcement signal r are utilized to change the weight of ASE. Then, the correct location and shape of the membership function is determined by the weight change. The triangular membership function (Figure 7) is initialized as the process information is available. As additional process information is available, the self-learning process does the following.

1. It gives appropriate output ($y = \mu_{BA}(\Delta B/\Delta A)$) according to the membership function.

2. When a failure signal (i.e., $y \neq 1$) is received, it adjusts the weight according to eqs 11 and 12. The membership function is reshaped according to eq 13 when Δw is available.

3. Repeat steps 1 and 2 until the correct diagnosis ($y = 1$) is achieved.

Therefore, the membership function is modified iteratively until the correct diagnosis, $\mu(\Delta B/\Delta A) = 1$, is achieved as shown in Figure 8. Typically, it takes less then 10 iterations to converge.

It is clear that the ability of the ASE goes beyond this type of application. In the Boxes system (Barto et al., 1983), it searches for appropriate control action as the state feedback becomes available. The control system emits the control action and the performance is evaluated. When a failure occurs, the reinforcement learning is made and another action is taken. Since the result of each learning step is checked on-line, the speed of convergence (to a successful learning) is critical for the control applications. In diagnosis, the performance after each learning step can easily be evaluated (to check whether $y = 1$ or not). Therefore, the speed of convergence is less critical. However, it differs from the box system in that the ASE recognizes that the reinforcement learning is to *include* additional process information as another valid set of input.

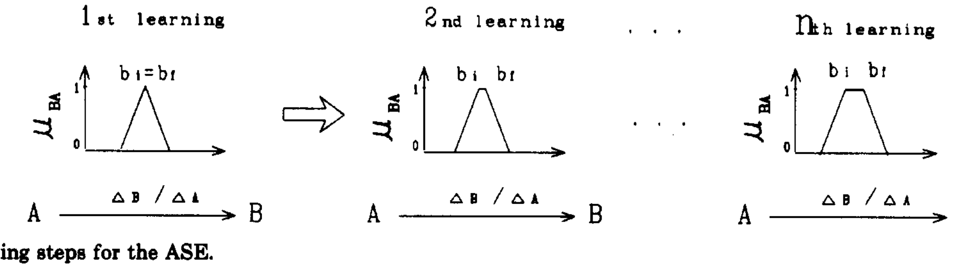

Figure 8. Learning steps for the ASE.

Table I. Fault Origins for the CSTR Example

symbol	fault origin
F_0	changes in the feed flow rate
C_{a0}	changes in the feed concentration
K_0	changes in the preexponential factor of rate constant
U	changes in the overall heat-transfer coefficient
T_{j0}	changes in the cooling water inlet temperature

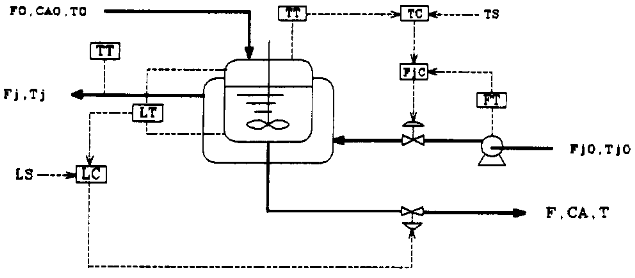

Figure 9. CSTR example.

Therefore, the staircase-like transfer function ($f(x)$ in eq 13) is used instead of the step function (Barto et al., 1983). Following this procedure, the self-learning qualitative/quantitative model *remembers* semiquantiative information at different operating conditions.

In the qualitative/quantitative fault model, there is an ASE associated with each branch for every fault origin. It provides the self-learning feature to the fault model such that the appropriate membership function is constructed to give correct the diagnosis (response) at different operating conditions.

4. Fault Diagnosis Systems

Two on-line diagnosis systems are investigated in this work. Both systems are associated with an artificial neural network (ANN), in some sense, e.g., either in the model structure or in the self-learning step. One system is the ANN with backpropagation learning which currently is the prototype of quantitative model-based diagnosis system (Watanabe et al., 1989; Venkatasubramanian et al., 1990; Ungar et al., 1990). This system can be viewed as a quantitative model which has the model structure of ANN. The other system utilizes the learning ability of the neural network to find the semiquantitative information. Specifically, it is a qualitative/quantitative model-based system with self-learning capability. A CSTR example (Figure 9) is employed to show the similarity and difference between these two systems. Before building any diagnosis system, the fault origins and process measurements have to be identified. For the CSTR example, the fault origins are listed in Table I. These faults include load changes (F_0, C_{a0}, and T_{j0}) and performance deterioration (U and K_0). The process measurements are temperatures, flow rates, and control signals as shown in Table II.

4.1. ANN with Backpropagation Learning. The inputs and outputs of the ANN with backpropagation learning are process measurements and fault origins, respectively. Once the process measurements are determined (Table II), the inputs to the ANN are obtained from the plant data or the results of computer simulations. In this work the quasi-steady-state results of the process simulator are employed to train the network. Typically, these variables are expressed in a dimensionless form.

Table II. On-Line Measurements for the CSTR

symbol	measured variable
T	reactor temperature
T_j	cooling water outlet temperature
F_j	cooling water flow rate
T_c	temperature controller output
F_{jc}	cooling water flow controller output
L_c	reactor level controller output

Table III. ANN Target Output Pattern

fault	target output pattern							
F_0^-	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
F_0^+	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
C_{a0}^-	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
C_{a0}^+	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
K_0^-	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
T_{j0}^-	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
T_{j0}^+	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
F_0^-	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
normal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

$$x = \frac{\text{measured } x - \text{nominal } x}{\text{nominal } x} \tag{14}$$

The target output patterns are determined from the fault origins with positive and/or negative deviations (Table III). In this work, there are five inputs and eight outputs in the ANN. In Table III, the value of "1" stands for a faulty state and "0" stands for normal operation. With input and output patterns available, the GDR is employed to supervise the learning of the network until the actual output patterns are close to the target output patterns within a threshold value.

Before the training procedure begins, several importance parameters, such as the number of elements in each layer and learning constants, have to be determined. As noted earlier, the number of elements in the input and output layers is chosen according to the measured variables and fault origins. However, there is no exact method to determine the number of elements in the hidden layer and learning constants. Generally, adquate values of these parameters are determined by trial and error (Watanabe et al., 1989; Venkatasubramanian et al., 1990).

4.2. Qualitative/Quantitative Model with Reinforcement Learning. The qualitative part of the model has the structure of a signed directed graph (SDG) which describes the causal effect between process variables. All the nodes, except the initial node (the fault origin), in a SDG are process measurements (Figure 10). The quantitative part of the model is formed using the membership function of fuzzy set theory as described in detail by Yu and Lee (1991). In this work, the semiquantitative knowledge is constructed via reinforcement learning. For the diagnostic system, the qualitative/quantitative models are constructed for both the steady state and the transient state. The dynamic responses of a faulty state are used to train the model for the diagnosis during the transient. When the changes of the process variables are less than

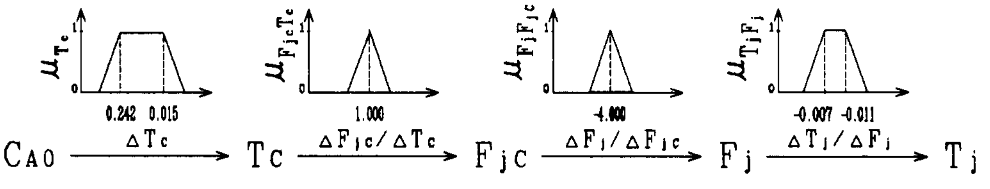

Figure 10. Qualitative/quantitative model from self-learning for the fault origin: a negative deviation in C_{a0}.

$$F_0 \xrightarrow{-} L_c \xrightarrow{+} T_c \xrightarrow{+} F_{jc} \xrightarrow{-} F_j \xrightarrow{-} T_j$$

$$C_{A0} \xrightarrow{-} T_c \xrightarrow{+} F_{jc} \xrightarrow{-} F_j \xrightarrow{-} T_j$$

$$K_0 \xrightarrow{-} T_c \xrightarrow{+} F_{jc} \xrightarrow{-} F_j \xrightarrow{-} T_j$$

$$U \xrightarrow{+} T_c \xrightarrow{+} F_{jc} \xrightarrow{-} F_j \xrightarrow{-} T_j$$

$$T_{j0} \xrightarrow{-} T_c \xrightarrow{+} F_{jc} \xrightarrow{-} F_j$$

Figure 11. Qualitative models (SDG) for all the fault origins.

5% between sampling instances, the steady state (or quasi steady state) is recognized and the corresponding process measurements are employed to train the model for the diagnosis at steady state. Therefore, these two qualitative/quantitative models handle the transient and steady-state responses separately.

The rule writing for the qualitative/quantitative model is similar to that of the qualitative model (Chang and Yu, 1990). That is, the degree of consistency for each fault propagation pathway is checked (Yu and Lee, 1991). For the CSTR example (Figure 9), the self-learned system results in the qualitative/quantitative model (Figure 10) for a negative deviation in C_{a0}. The membership functions in Figure 10 are obtained via reinforcement learning using several sets of quasi-steady-state information (−10, −30 and −60% deviations in C_{a0}. The rule can be written as

$$\mu_{C_{a0}^-} = \min\,[\mu_{T_c}(\Delta T_c),\ \mu_{F_{jc}T_c}(\Delta F_{jc}/\Delta T_c),\ \mu_{F_jF_{jc}}(\Delta F_j/\Delta F_{jc}),\ \mu_{T_jF_j}(\Delta T_j/\Delta F_j)] \quad (15)$$

where $\mu_{C_{a0}^-}$ is the truth value for C_{a0} going through a negative change, the "min" operator taking the smallest value in the bracket. Here, $\mu_{T_c}(\Delta T_c)$, $\mu_{F_{jc}T_c}(\Delta F_{jc}/\Delta T_c)$, etc. are the degree of consistency for each branch in Figure 10. Similarly, rules can also be written for the positive deviations in C_{a0} and other fault origins.

5. Applications

A CSTR example (Chang and Yu, 1990; Yu and Lee, 1991) is used to illustrate the performance of the qualitative/quantitative model with reinforcement learning. The proposed approach is compared to the ANN diagnostic system with backpropagation learning.

5.1. Process. In this example, an irreversible and exothermic reaction is carried out in a perfectly mixed CSTR as shown Figure 9. Parameter values are taken from Luyben (1990). Eight faulty states with both the negative and positive deviations (Table I) are to be diagnosed. Table II shows the measured variables in this study. Basically, these measurements can be obtained with little difficulty. Notice that the concentration of the reactant A in the reactor, C_a, is not included. The consideration is a practical one: on-line composition measurements often are not available in reaction units. This example poses a difficult diagnosis problem. If only qualitative process measurements are available, the faults C_{a0}, K_0, and U (the changes in feed concentration, rate constant, and overall heat-transfer coefficient) are indistinguishable as shown in the SDG's of the fault origins (Figure 11). However, if quantitative process measurements are available, only two faults C_{a0} and K_0 are not distinguishable (Figure 12). Figure 12 shows that the steady-state gains between measured variables which are derived from the linearized process model. Therefore, from the linear analysis, fault in C_{a0} and K_0 cannot be separated. However, for a range of deviations in C_{a0} and K_0, the steady-state gains between the nodes F_j–T_j are not quite the same as shown in Figure 13. That is, it is possible to distinguish these two faults from a nonlinear analysis. It should be pointed out that the magnitudes of the faults of interest are between 10% and 60%. In this work, the sampling time for the diagnostic system is 3 min. That is, the process measurements are sampled every 3 min and the diagnosis is made right after. Therefore, the diagnosis results can be shown on the CRT of the process control computer as the "diagnosis" trend.

$$F_0 \xrightarrow{-6.6667} L_c \xrightarrow{0.1033} T_c \xrightarrow{1.0000} F_{jc} \xrightarrow{-4.0000} F_j \xrightarrow{-0.0083} T_j$$

$$C_{A0} \xrightarrow{-0.4122} T_c \xrightarrow{1.0000} F_{jc} \xrightarrow{-4.0000} F_j \xrightarrow{-0.0083} T_j$$

$$K_0 \xrightarrow{-0.2022} T_c \xrightarrow{1.0000} F_{jc} \xrightarrow{-4.0000} F_j \xrightarrow{-0.0083} T_j$$

$$U \xrightarrow{0.0207} T_c \xrightarrow{1.0000} F_{jc} \xrightarrow{-4.0000} F_j \xrightarrow{-0.1090} T_j$$

$$T_{j0} \xrightarrow{-0.1392} T_c \xrightarrow{1.0000} F_{jc} \xrightarrow{-4.0000} F_j$$

Figure 12. Steady-state gains for all fault origins on branch from a linearized model.

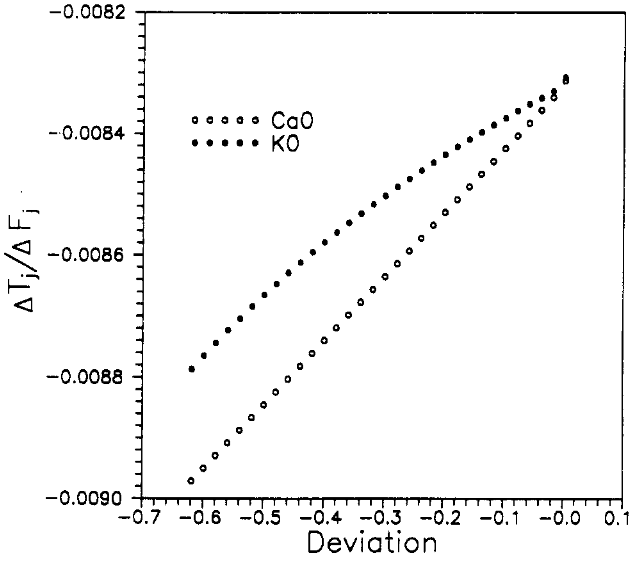

Figure 13. Steady-state gains for the branch "$F_j \rightarrow T_j$" for different degrees of negative deviations in C_{a0} and K_0.

5.2. ANN Diagnostic Systems. Two ANN's with backpropagation learning are constructed and tested for this CSTR example. These two ANN's differ in the numbers of input patterns and the structure (the numbers of hidden layers).

In the first ANN, two input patterns, 10% and 30% deviations in the fault origins, are employed for each fault origin. The process measurements are obtained from process simulation when the responses approach steady-state (e.g., at 3.5 h after the fault initiates). The fault origins and target output patterns are shown in Table III. The input variables are the on-line measurements including L_c, T_c, T, F_j, and T_j (Table II). After a period of trial and error, a three-layered neural network is chosen. The numbers of the elements in the input, hidden, and output layers are 5, 10, and 8, respectively (Figure 14). This neural network is called ANN(I) hereafter. The

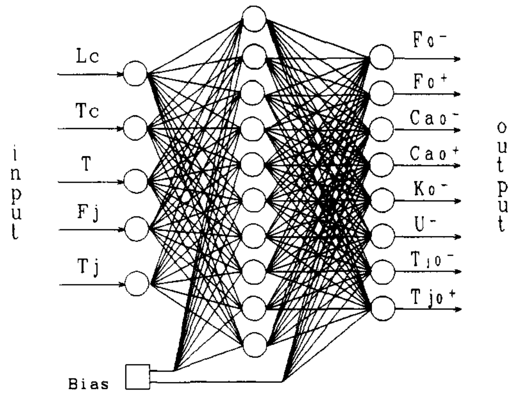

Figure 14. ANN diagnostic system (ANN(I)).

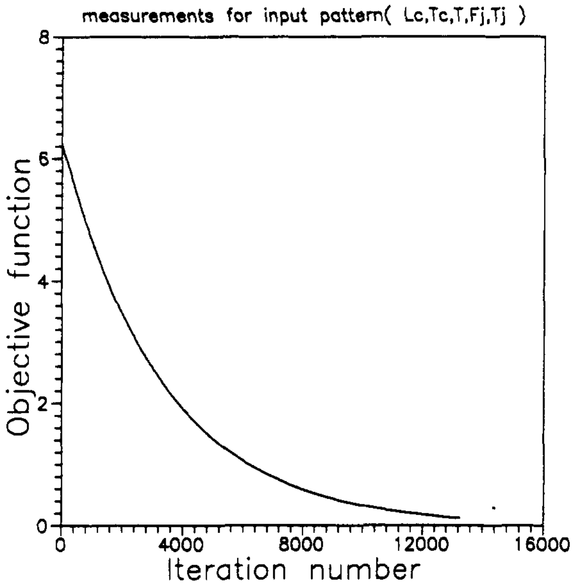

Figure 15. Convergence of the objective function for ANN(I).

learning rate $\eta = 0.6$ and momentum term $\beta = 0.9$ are used in ANN(I). It takes approximately 13 000 iterations to converge to the criterion $E < 0.08$. The response of learning is shown in Figure 15.

Once the ANN(I) is constructed and the training is successful (satisfying the convergence criterion), the diagnostic system is tested on-line. For the trained patterns, e.g., −30% deviation in C_{a0}, ANN(I) gives a perfect result. Figure 16 shows that ANN(I) identifies the fault origin (C_{a0}^-) correctly 1 h after the fault starts. Furthermore, there is no spurious solution in this case. That is, ANN(I) does a superb job in identifying the fault origin for the trained pattern. Note that only steady-state information is used in the training step. Unfortunately, ANN(I) gives an erroneous solution (fails to identify the true fault origin) as interpolation and/or extrapolation between input patterns is required (Figures 17 and 18). Figure 17 shows that when C_{a0} goes through a range of negative deviation (−10% to −60%), ANN(I) misses the true fault origin (C_{a0}^-) for ΔC_{a0} between −12% and −25%. In this case, ANN(I) finds the fault origin K_0^- instead (Hsu, 1991). The results shown here reveal a serious problem associated with ANN(I): the only solution given by ANN(I) is erroneous. Similar results can also be found for a range of deviations in K_0 (Figure 18). Again, ANN(I) gives erroneous solutions (finds C_{a0}^- instead) for two ranges of ΔK_0 as shown in Figure 18.

In order to improve the performance of ANN(I), an attempt is made by including another input pattern (60% deviation in the fault origin) to train the ANN model.

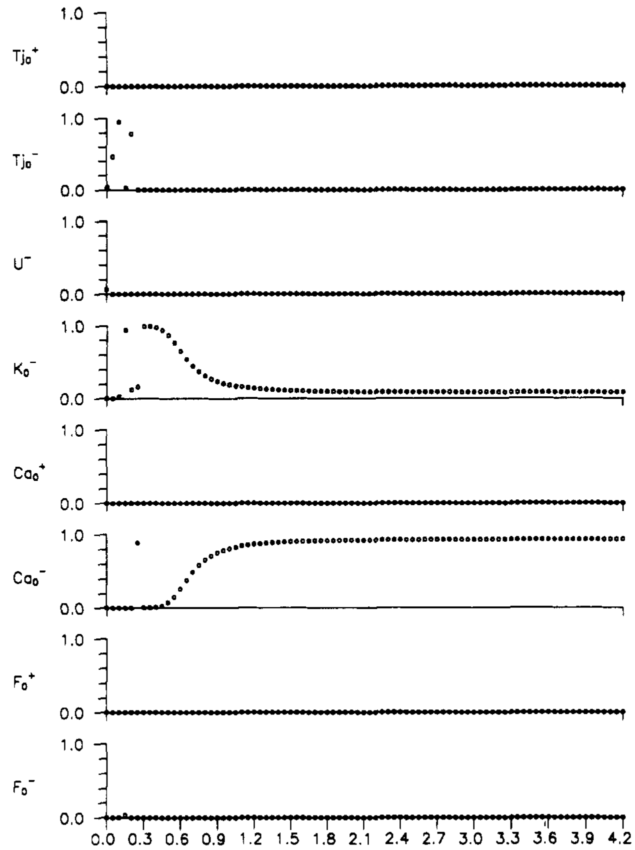

Figure 16. Diagnosis results of ANN(I) for a 30% negative deviation in C_{a0} (a trained input pattern).

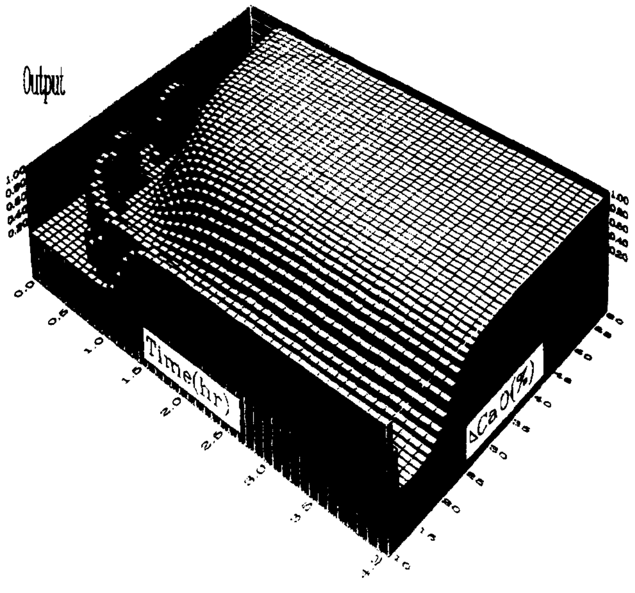

Figure 17. Diagnosis results of ANN(I) for a range of negative deviations in C_{a0} (−10% to −60%).

However, the three-layered neural network (Figure 14) fails to converge. A four-layered neural network is tested. After some trials and errors, the ANN with the numbers of elements of 5, 15, 15, and 8 in the input layer, hidden layer 1, hidden layer 2, and the output layer is chosen. The learning rate η and momentum term β are 0.1 and 0.9, respectively. This neural network is called ANN(II) hereafter. The differences between ANN(I) and ANN(II) are ANN(II) is a four-layered neural network and three input patterns for each fault origin are employed in ANN(II).

Upon diagnosis, ANN(II) also performs perfectly for the trained patterns (10%, 30%, and 60% deviation (Hsu,

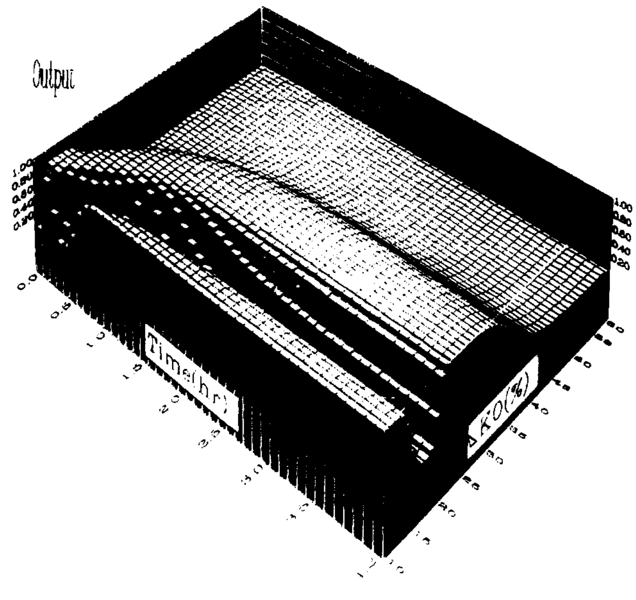

Figure 18. Diagnosis results of ANN(I) for a range of negative deviations in K_0 (−10% to −60%).

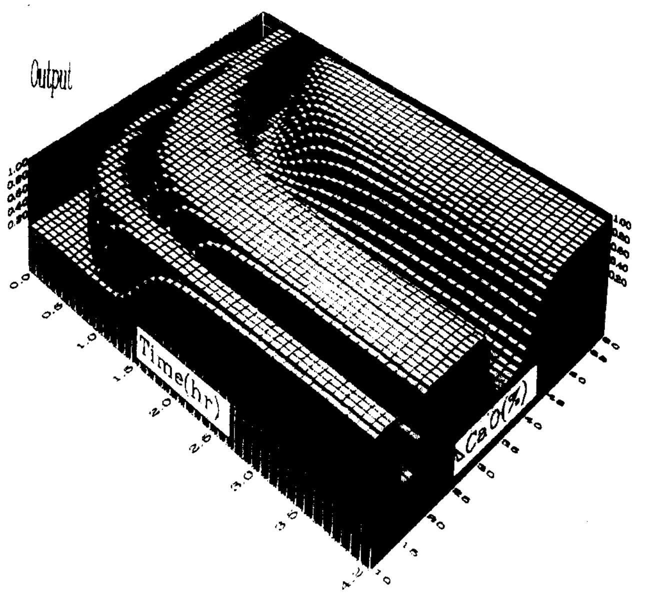

Figure 19. Diagnosis results of ANN(II) for a range of negative deviations in C_{a0} (−10% to −60%).

1991)). Unfortunately, ANN(II) also gives erroneous solutions for the ranges of deviations in C_{a0} and K_0 (Figures 19 and 20). Furthermore, for the fault origin C_{a0}^-, diagnostic results simply deteriorate as shown in Figures 17 and 19. Apparently, little improvement is achieved by including one more input pattern and one more hidden layer. The characteristics of ANN(I) or ANN(II) shown here certainly limits the applicability of ANN in any practical situation for this type of process. Note that the results shown here do not imply that backpropagation ANN is not suitable for fault diagnosis in all cases. The CSTR example poses a very difficult diagnosis problem as pointed out earlier.

5.3. Qualitative/Quantitative Diagnostic Systems. The qualitative part of the fault model is constructed first followed by the self-learning of the semiquantitative knowledge. In the self-learning phase, 10%, 30%, and 60% deviations in each fault origin are used to shape the semiquantitative process knowledge using ASE. Let us consider the case of C_{a0} going through negative changes as an example to illustrate the learning process. The steady-state qualitative/quantitative model can be found via reinforcement learning. The process measurements (T_c, F_{jc}, F_j, and T_j) at quasi steady state (1.5 and 4 h after the occurrence of the fault) are recorded and converted to steady-state gains for the training of the corresponding branches. Therefore, there are six data points for a single branch. For the branch between F_j and T_j (Figure 10), the gains range from −0.0067 to −0.0087. Initially, one has no a priori knowledge about the location of the membership function. When the information comes in ($\Delta T_j/\Delta F_j$ = −0.0067), a triangular-shaped membership function is formed with the apex located at −0.0067 and it decreases linearly to zero for ±30% deviations from the apex (μ becomes 0 at −0.0045 and −0.0089). When the second set of data comes in ($\Delta T_j/\Delta F_j$ = −0.0087), an unsatisfactory diagnosis result is obtained, i.e., $\mu_{T_jF_j}(\Delta T_j/\Delta F_j) \approx 0$, and the reinforcement signal is activated (r = 1). The ASE adjusts the membership function in the following way (Figure 7): (1) calculating $e(t+1)$ from eq 12, (2) finding Δw using eq 11, and (3) changing the membership function according to eq 13 (initially $b_i = b_f$ = −0.0067). These three steps are repeated until a satisfactory diagnosis results (i.e., $\mu_{T_jF_j}(-0.0087) = 1$). In this example, it takes two iterations to converge and the resulting semiquantitative model is shown in Figure 10 (the F_j–T_j branch). Since the other four gains falls between −0.0067 and −0.0087, satisfactory diagnostic results are produced and the ASE is not activated. This procedure is repeated for all branches with all fault origins. In this work, the learning rate of 1 and the trace decay ratio of 0.9 are used throughout. A typical qualitative/quantitative model constructed from reinforcement learning is similar to the one shown in Figure 10.

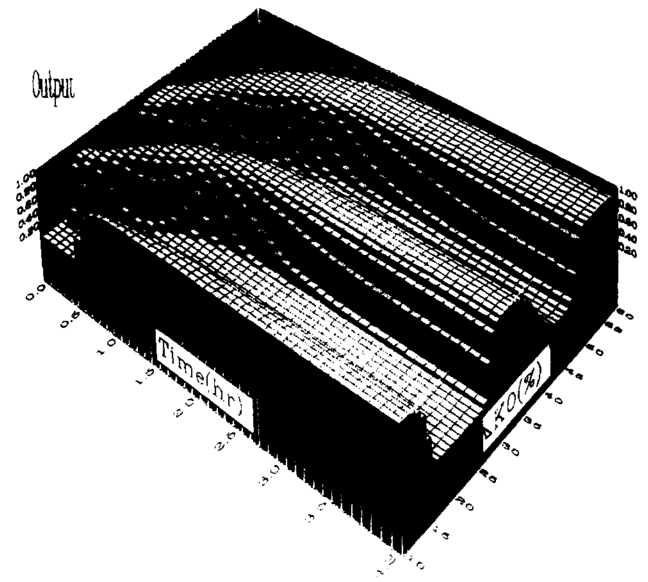

Figure 20. Diagnosis results of ANN(II) for a range of negative deviations in K_0 (−10% to −60%).

In the diagnosis phase, the diagnostic system is tested against each fault origin with a range of deviations. The procedure for fault diagnosis is exactly the same as that of Yu and Lee (1991). Unlike the ANN diagnostic systems, e.g., ANN(I) and ANN(II), the proposed diagnostic system does not give erroneous solutions (Figure 21). Figure 21 shows that the qualitative/quantitative model-based diagnostic finds the true fault origin for a range of negative deviation (−10% to −60%) in C_{a0}. However, it results in spurious solutions as shown in a −20% deviation of C_{a0} (Figure 22) or a −20% deviation of K_0 (Figure 23). In both cases, it finds C_{a0}^- and K_0^- as the fault origins. Despite

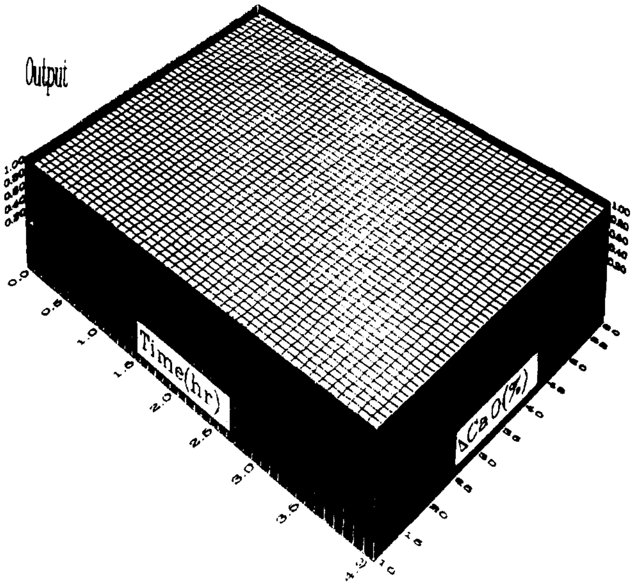

Figure 21. Diagnosis results for the qualitative/quantitative model for a range of negative deviations in C_{a0} (−10% to −60%).

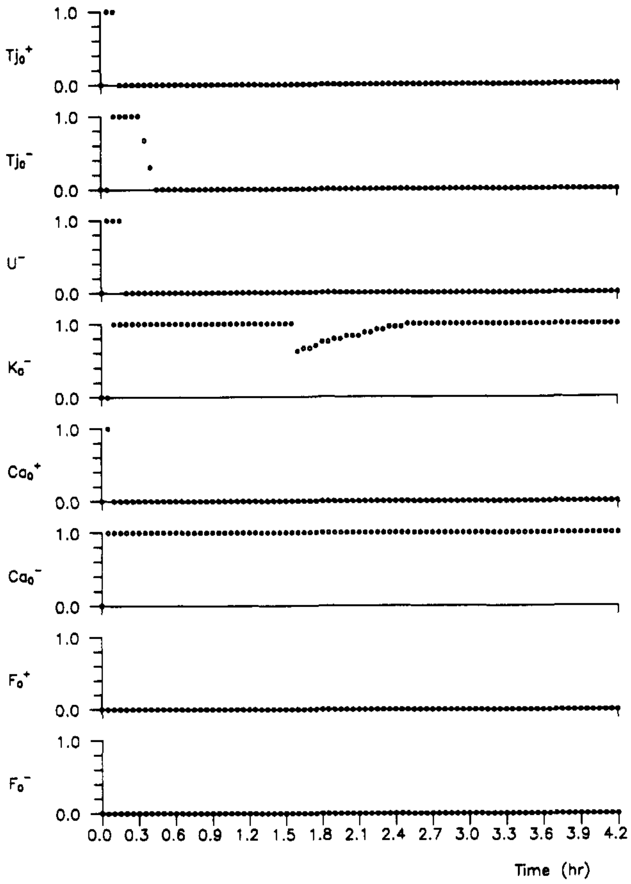

Figure 22. Diagnosis results for the qualitative/quantitative model for a 20% negative deviation in C_{a0}.

the possibility of giving spurious solutions, the qualitative/quantitative diagnostic system shows a very desirable characteristic: it does not give erroneous interpretations. The reason is that the membership function-based qualitative/quantitative model adapts to new (additional) information by including it (instead of changing to a new crisp point as quantitative models do). This clearly shows the flexibility of the qualitative/quantitative model (as opposed to the rigidity of quantitative models). In summary, the qualitative/quantitative model-based diagnosis system has the advantage over the strictly quantitative model-based system (e.g., ANN(I) and ANN(II)), since it did not produce erroneous solutions. It also has the advantage over the strictly qualitative model-based system (e.g., SDG) for producing less spurious solutions. Furthermore, the semiquantitative information is self-learned via the ASE which requires little engineering effort.

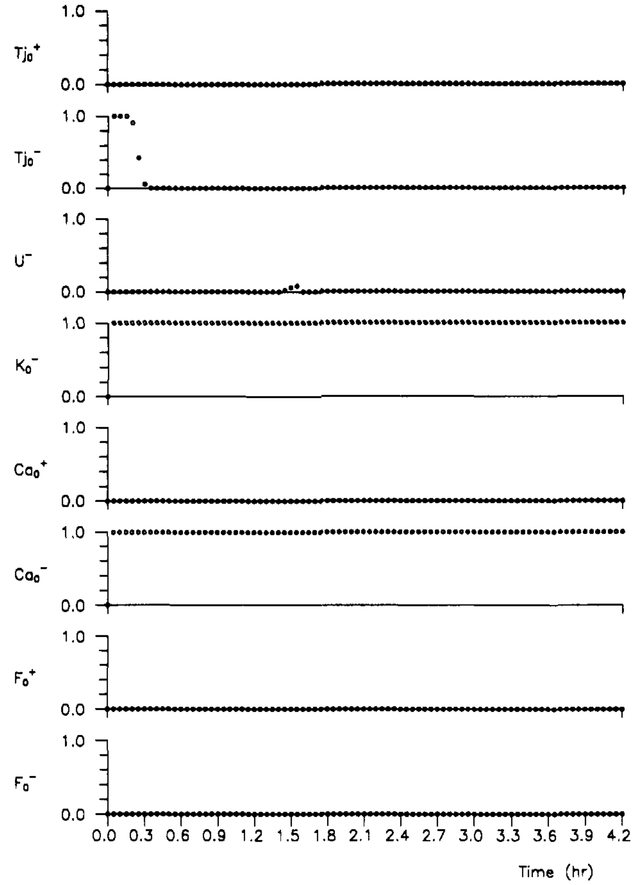

Figure 23. Diagnosis results for the qualitative/quantitative model for a 20% negative deviation in K_0.

5.4. Discussion. Despite the fact that both diagnostic systems have the "learning" feature, the performance between the two is quite different. The ANN's (ANN(I) and ANN(II)) try to learn to reproduce the trained input patterns. However, the backpropagation learning of ANN ignores an engineering fact that the gains between T_j–F_j are almost the same for the fault origins C_{a0} and K_0. Without taking this fact into consideration, GDR simply tries to converge to the target output patterns by assigning one range of $\Delta T_j/\Delta F_j$ to C_{a0} and another range of $\Delta T_j/\Delta F_j$ to K_0 (e.g., Figure 13) according to the input patterns supplied. Therefore, the ANN diagnostic system does not catch the global view; e.g., $\Delta T_j/\Delta F_j$ for both fault origins can take any possible value between −0.0083 and −0.0088. This, subsequently, leads to erroneous solutions as shown in the diagnostic results.

The self-learning feature of the qualitative/quantitative model, on the other hand, is confined to the semiquantitative part of the process knowledge. That is, we keep the structure of the model unchanged and modify the more rigid quantitative information when needed. Furthermore, the reinforcement learning modifies the membership function by *including* the new process data instead of adapting to the new information. "Learning" under these guidelines is not likely to give erroneous solutions when we interpolate between the trained patterns. The diagnostic results also confirm this.

It should be emphasized that the CSTR example studied poses a quite difficult diagnosis problem. This difficulty results from the selection of the process measurements and faults to be diagnosed. For example, if C_a is measurable, all the fault origins can be correctly identified with only qualitative values of process measurements (Hsu, 1991). This implies that, in many occasions, the difficulty in diagnosis arises from the *selected measurements*, not from the process itself. Therefore, selection of the appropriate measurements for fault diagnosis can simplify the effort in fault diagnosis.

6. Conclusion

A self-learning feature is proposed for the qualitative/quantitative model-based diagnostic system. Based on the reinforcement learning of neural network, a single neuron (ASE) is used to shape the semiquantitative part of the process knowledge. This provides the self-learning ability to a diagnostic system in a transparent manner. Comparisons are made between the qualitative/quantitative model with reinforcement learning and the ANN with backpropagation learning. Simulation results show that the proposed self-learning diagnostic system is not only transparent in analyses but superior in performance (as far as the completeness is concerned). More importantly, the self-learning feature makes the qualitative/quantitative model-based diagnostic system attractive in practical applications, since it requires much less engineering effort.

Acknowledgment

This work is supported by the National Science Council of the ROC under Grant NSC 81-0414-P-011-04-B.

Nomenclature

ANN = artificial neural network
ASE = associative search element
b_i = initial value in the membership function (the upper left corner of the trapezoid)
b_f = final value in the membership function (the upper right corner of the trapezoid)
C_a = concentration of reactant A
C_{a0} = feed concentration of reactant A
d_i = desired output value
e = eligibility
E = objective function
$f(\cdot)$ = transfer function in the neural network
$f'(\cdot)$ = derivative of $f(\cdot)$
F_0 = feed flow rate
F_j = cooling water flow rate
F_{jc} = cooling water flow controller output
GDR = generalized delta rule
K_0 = preexponential factor of the rate constant
L_c = reactor level control output
min $(\cdot)$ = minimum value of $(\cdot)$
r = reinforcement
SDG = signed directed graph
T = reactor temperature
T_c = temperature controller output
T_{j0} = cooling water inlet temperature
$t + 1$ = $(t + 1)$th iteration
U = overall heat-transfer coefficient
w = weight in ASE
w_i = weight of input pathway in the Boxes system
w_{ij} = weight between the ith element of the input layer and the jth element of the upper layer
Δw = weight change in ASE
Δw_{ij} = change of weight between iterations
x = input to ASE
x_i = ith input of backpropagation ANN
y = output of ASE
y_i = ith output of backpropagation ANN

Greek Symbols

α = learning rate in ASE
β = momentum term in backpropagation ANN
δ = trace decay rate in ASE
δ_j = error term in backpropagation ANN
η = learning rate in backpropagation ANN
θ_j = bias in backpropagation ANN
μ_A = membership function of A

Literature Cited

Barto, A. G.; Sutton, R. S.; Anderson, C. W. Neuronlike Adaptive Elements That Can Solve Difficult Control Problems. *IEEE Trans. Syst., Man Cybern.* **1983**, *SMC-13*, 834–847.

Chang, C. C.; Yu, C. C. On-Line Faults Diagnosis Using Signed Directed Graph. *Ind. Eng. Chem. Res.* **1990**, *29*, 1290–1299.

Cheung, J. T. Y.; Stephanopoulos, G. Representation of Process Trends—I A Formal Representation Framework. *Comput. Chem. Eng.* **1990a**, *14*, 495–510.

Cheung, J. T. Y.; Stephanopoulos, G. Representation of Process Trends—II The Problem of Scale and Qualitative Scaling. *Comput. Chem. Eng.* **1990b**, *14*, 511–539.

Frank, P. M. Fault Diagnosis in Dyanmic Systems Using Analytical and Knowledge-Based Redundancy—A Survey and Some New Results. *Automatica* **1990**, *26*, 459–474.

Himmeblau, D. M. *Fault Diagnosis and Detection in Chemical and Petrochemical Processes*; Elsevier: Amsterdam, 1978; p 1.

Hsu, Y. Y. Automatic Fault Diagnosis Systems: Associative Reinforcement Learning. M.S. Thesis, National Taiwan Institute of Technology, Taipei, 1991 (in Chinese).

Isermann, R. Process Fault Detection Based on Modeling and Estimation Method—A Survey. *Automatica* **1984**, *20*, 387–404.

Kramer, M. A.; Palowitch, B. L., Jr. A Rule-Based Approach Diagnosis Using the Signed Directed Graph. *AIChE J.* **1987**, *33*, 1067–1087.

Kuipers, B. The Qualitative Calculus is Sound but Incomplete: A Reply to Peter Struss. *Artif. Intell. Eng.* **1988**, *3*, 170–173.

Lee, C. C. A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Net Concepts. *Int. J. Intell. Syst.* **1991**, *6*, 71–93.

Lippmann, R. P. An Introduction to Computing Neural Nets. *IEEE, ASSP Mag.* **1987**, *April*, 4–22.

Luyben, W. L. Process Modeling, Simulation and Control for Chemical Engineers, 2nd ed.; McGraw-Hill: New York, NY, 1990; p 124.

Petti, T. F.; Klein, J.; Dhurjati, P. S. Diagnostic Model Processor: Using Deep Knowledge for Fault Diagnosis. *AIChE J.* **1990**, *36*, 565–575.

Prasad, P. R.; Davis, J. F. A Framework for Implementing On-Line Diagnostic Advisory Systems in Continuous Process Operations. AIChE Annual Meeting, Nov 17–22, 1991, Los Angeles.

Rengaswamy, R.; Venkatasubramanian, V. An Integrated Framework for Process Monitoring, Diagnosis, and Control Using Knowledge-Based Systems and Neural Network. IFAC Symposium on On-Line Fault Detection and Supervision in Chemical Process Industries, April 22–24, 1992, Newark, DE.

Rumelhart, D. E.; McClelland, J. L., Eds. *Parallel Distributed Processing*; MIT Press: Cambridge, MA, 1986; p 324.

Ungar, L. H.; Powell, B. A.; Kamens, S. N. Adaptive Networks for Fault Diagnosis and Process Control. *Comput. Chem. Eng.* **1990**, *14*, 561–572.

Venkatasubramanian, V.; Vaidyanathan, R.; Yamamoto, Y. Process Fault Detection and Diagnosis Using Neural Networks—I. Steady-State Process. *Comput. Chem. Eng.* **1990**, *14*, 699–712.

Watanabe, K.; Matsuura, I.; Abe, M.; Kubota, M.; Himmelblau, D. M. Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks. *AIChE J.* **1989**, *35*, 1803–1812.

Willsky, A. S. A Survey of Design Methods for Failure Detection in Dynamic Systems. *Automatica* **1976**, *12*, 601–611.

Yu, C. C.; Lee, C. Fault Diagnosis Based on Qualitative/Quantitative Process Knowledge. *AIChE J.* **1991**, *37*, 617–628.

Received for review July 25, 1991
Revised manuscript received December 12, 1991
Accepted March 23, 1992