Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks

Artificial neural networks have capacity to learn and store information about process faults via associative memory, and thus have an associative diagnostic ability with respect to faults that occur in a process. Knowledge of the faults to be learned by the network evolves from sets of data, namely values of steady-state process variables collected under normal operating condition and those collected under faulty conditions, together with information about the degree of the faults and their causes.

Here, we describe how to apply artificial neural networks to fault diagnosis. A suitable two-stage multilayer neural network is proposed as the network to be used for diagnosis. The first stage of the network discriminates between the causes of faults when fed the noisy process measurements. Once the fault is identified, the second stage of the network estimates the degree of the fault. Thus, the diagnosis of incipient faults becomes possible.

Kajiro Watanabe
Ichiro Matsuura
Masahiro Abe
Makoto Kubota
Department of Instrument and Control Engineering
Hosei University
Tokyo 184, Japan

D. M. Himmelblau
Chemical Engineering Department
University of Texas
Austin, TX 78712

Introduction

As time passes, the performance of chemical process equipment gradually degrades due to deterioration of process components. In addition, ambient disturbances engender process upsets. Both factors lead to values of process variables at variance with those expected under normal operating conditions. Deviation of a variable from its normal value in the incipient stage of the occurrence of a fault is slight and may well be concealed by the operation of feedback control. If the fault is not corrected, the process in some instances could incur a catastrophic event; a fatal accident might result from a malfunction of equipment (Himmelblau, 1978; Watanabe, 1983a). Diagnosis of faults, which are in an incipient stage but that might lead to a serious situation in the near future, is an essential element of plant operations.

From the viewpoint of preventive maintenance, a regular check-up of equipment components by dismantling them when the process is shut down, is one method that assists in accurate diagnosis. But between regular checks-ups, on-line quantitative diagnosis of the operating process is desirable. We treat the problem of the quantitative on-line diagnosis of incipient faults from noisy process measurements using just a few variables from the operating process and without requiring any special process inputs or perturbations.

Various quantitative methods to diagnose incipient faults can be found in the literature. For example, Beard (1971) presented a filtering (full-order Luenberger observer) technique that generated estimation errors for the state variables with patterns corresponding to known failure modes, a method which is recommended by Willsky (1976) in his review paper. Watanabe and Himmelblau (1983b, c, 1984) applied the extended Kalman filter and/or nonlinear estimator to identify process parameters indicative of process faults caused by deterioration of components. The difficulty with fault detection via a state space approach is that the technique involves extensive process modeling and a heavy calculational load for identification or estimation. Process modeling for fault diagnosis itself can be quite a difficult job because errors in the model can (a) be interpreted as faults, thus yielding false alarms, or (b) prevent faults from being detected when they occur.

Rule-based expert systems with Boolean (binary) reasoning or those with non-Boolean (for example, fuzzy sets theory) reasoning are one of the practical ways to represent knowledge of faults; they can be effectively applied to fault diagnosis.

Binary reasoning can only provide qualitative diagnosis whereas systems with non-Boolean reasoning can provide quantitative diagnosis. Kramer (1987) demonstrated that a non-Boolean expert system yielded stable and quite sensitive diagnoses in the presence of noise. Further, he described a method to narrow the diagnostic focus using function decomposition (Kramer, 1988). Fault diagnosis via rule-based expert systems, however, does require a database of rules about the faults. The rules must be generated by process experts, operators, and engineers who analyze the historical causes and modes of equipment failure. The accuracy of a diagnosis depends on how rich and accurate the database is. Construction of the database itself is quite time-consuming (and expensive), and is as difficult a job as process modeling.

Artificial neural networks (McClelland and Rumelhart, 1986; Aso, 1988) are an alternative to represent knowledge about faults. A neural network can autonomously store knowledge by learning from historical fault information and has the characteristic of associative memory. Information about faults can be learned by training the network on a set of data such as the values of steady-state process variables for normal conditions and those for identified faulty conditions. If data cannot be found in the past daily reports of process maintenance and data logs, the necessary data can be collected from a designated process quality control program designed to identify faults. Because a network can filter out noise and classify faults, and because the information needed to train the network can be easily collected, we suggest that artificial neural networks can provide fully-automatic, quantitative, stable, highly-sensitive and economical diagnosis of faults even in the presence of noise. Gallant (1987) described automated generation of connectionist expert systems. Ungar and Powell (1988) described adaptive networks for fault diagnosis. Hoskins and Himmelblau (1988) demonstrated that artificial neural networks have considerable potential when applied to the diagnosis of chemical processes.

This paper describes how to effectively apply artificial neural networks to fault diagnosis problems in a chemical reactor. We present a new architecture comprising a two-stage multilayer neural network and use it to discriminate among the causes of faults as well as to estimate the degree of deterioration represented by the faults in the presence of noisy measurements.

Artificial Neural Networks

In this section we briefly summarize the characteristics of artificial neural networks (refer to Figure 1). The roots of these ideas lie in simplified explanations of the functioning of human and animal brains.

Artificial neuron

An artificial neuron is a simple processing element that serves as a transfer function mapping a multidimensional input

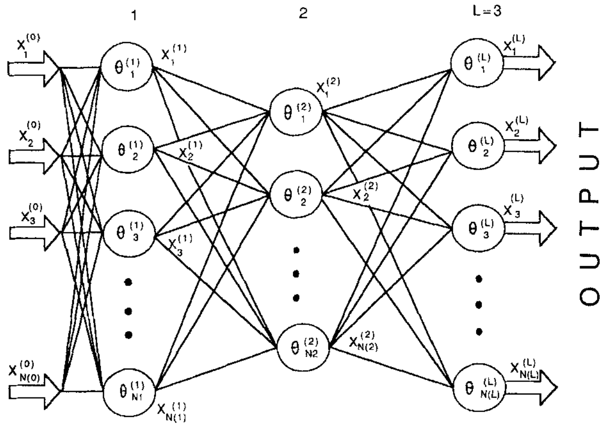

Figure 1. Multilayer neural network.

received from other artificial neurons or external stimuli to a one-dimensional output which is distributed to other artificial neurons through weighted connections. The transfer function of the jth artificial neuron in ℓth layer in Figure 1 is specified by a sigmoidal function

$$x_j^{(\ell)} = \frac{1}{1 + e^{-u_j^{(\ell)}}} \tag{1a}$$

with $u_j^{(\ell)}$ being usually (but not always) a linear sum of the weighted connection strengths being fed to the node plus a threshold

$$u_j^{(\ell)} = \left\{ \sum_{i=1}^{N-1} w_{ji}^{(\ell)} x_i^{(\ell-1)} + \theta_j^{(\ell)} \right\} \tag{1b}$$

where variable $x_j^{(\ell)}$ is the output or the activity level of jth node in the ℓth layer and variable $x_i^{(\ell-1)}$ is the output of ith node in the $(\ell - 1)$th layer, when the input pattern p is fed to the network; $\theta_j^{(\ell)}$ is the threshold of jth node in ℓth layer.

Network topology

Figure 1 shows a standard multilayer feedforward artificial neural network with one or more so-called hidden layers ("hidden" because such layers do not communicate directly with the external environment). The arcs that connect the artificial neurons are unidirectional feedforward connections. Let number of neurons in layer ℓ be N_ℓ. The arc from ith node in the $(\ell - 1)$th layer to jth node in ℓth layer has an associative weight $w_{ji}^{(\ell)}$ which multiplies the signal from ith node in $(\ell - 1)$th layer. Knowledge in artificial neural networks is distributed among the connections and the weights $w_{ji}^{(\ell)}$ and not stored at a single computer address. For fault detection each node in the output layer would represent a particular fault; one additional node would represent normal operating conditions.

Processing in a multilayer artificial neural network

Each node in the input layer receives input from an external stimulus that is either scaled prior to introduction into the respective input node or scaled by the node. The output of each node in the input layer is passed on to all the nodes in the next layer. Each artificial neuron in the next layer computes an output (activity level) that is a function of its inputs. The computations within a layer are asynchronous and thus may be performed in parallel. The output of one node is distributed to all the other artificial neurons in the subsequent layer through the weighted connections. This arrangement is repeated in a feedforward manner until the output layer is reached. Thus, computations between layers are synchronous.

Learning (training)

Learning is nothing more than adjusting the weights associated with connections between the nodes of the network. An input vector of process measurements associated with a fault pattern is introduced into the input layer of the network. A corresponding output pattern composed of output layer node activities is calculated. An error is generated for each output node based on the difference from a target value for the node or a goal. For fault detection, an output node target would be either 0 (no fault) or 1 (a particular fault). The neural network learns a target output pattern by adjustment of the weights in the network; therefore, after a sequence of presentations of input vectors, the network generates the desired output pattern for its associated input measurement vector. To adjust the weights, we used the backpropagation procedure (McClelland and Rumelhart, 1988) in which the objective function

$$E_p = \frac{1}{2} \sum_{j=1}^{N_L} \{t_{pj}^{(L)} - x_{pj}^{(L)}\}^2 \tag{2a}$$

is minimized for a given input pattern p, where $t_{pj}^{(L)}$ is the target output (activity) of the jth node in the output layer for pattern p. The same input vector may be used periodically during the learning process.

Learning via backpropagation involves two phases. In the first phase, the inputs are propagated in a feedforwarded manner through the network to produce output values that are compared to the target values, resulting in the error signal for each of the output nodes. In the second phase, the errors are propagated backward through the network and used to adjust the weights. The error signals for the output layer are calculated first, and these error signals are used recursively to calculate the needed adjustments layer by layer until the weights for all of the connections are recalculated.

To train the network (complete the learning), we minimized

$$E = \sum_{p=1}^{N_p} E_p \tag{2b}$$

where N_p is the number of input vectors used in the learning process. If noise is added to deterministic input vectors, the learning takes longer, but the network is better able to generalize, i.e., properly classify input vectors not used in the learning process.

Pattern recognition (fault detection)

Once the weights on the connections are finally adjusted in the training phase, a new vector of sensor measurements can be sent to the input nodes of the network and classified. Very little computation time is needed for this step. The degree of misclassification that occurs is a function of how well the knowledge stored in the connections and weights can represent perturbations from the training set of data.

Simulated Faults in a Reactor

Consider a PI-feedback-controlled process, in which heptane is converted catalytically to toluene and which is operated near its steady-state operating conditions. No catastrophic and/or abrupt event occurs because any process fault is in the incipient stage and the effect of any fault is strongly regulated and alleviated by the controller. Thus, no change in reactor temperature occurs as a result of a fault. Figure 2 shows the process. Heptane stored in the tank is fed to the reactor through process pump 1. The catalytic reaction occurring in the reactor is:

$$C_7H_{16} \rightarrow C_7H_8 + 4H_2$$

The reaction rate is controlled by the temperature in the reactor. Steam supplied to the heat exchanger in the reactor is recycled to the inlet of the heater via recycle pump 2. The outlet valve in the heater is controlled by the PI controller. The controlled variable is the reaction temperature, and the controller regulates the temperature to the specified value. The lumped models of

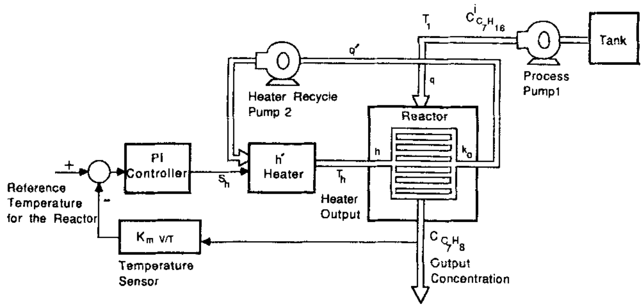

Figure 2. Process to be diagnosed.

the reactor, heater and controller that were used to simulate various faults are shown in the Appendix. We assumed that the heater control comes from an electronic circuit or a process computer, and we also assumed that the controller has the property of self-diagnosis and self-repair, and thus is always in a nonfaulty operating mode.

To illustrate the use of an artificial neural network in diagnosing faults, we categorized five possible causes of faults as follows:

Fault 1. Deterioration of catalyst performance, due to physical and/or chemical deterioration of the catalyst, leads to a decrease in the frequency factor for the catalyst, k_0, in the model.

Fault 2. Fouling of the heat exchanger surface in the reactor leads to a decrease of overall heat transfer coefficient, h, in the model.

Fault 3. Fouling of the heat exchanger surface in the heater leads to a decrease in overall heat transfer coefficient, h', in the model.

Fault 4. Partial plugging of the pipeline connected to pump 1 leads to a decrease in the volumetric flow rate, q, in the model.

Fault 5. Partial plugging of the pipeline connected to pump 2, leads to a decrease in the volumetric flow rate, q', in the model.

We discriminated among and/or diagnosed the existence of the above five faults from measurements of the outlet concentration of $C_{C_7H_8}$, the heater outlet temperature T_h, and the controller output signal s_h.

Knowledge about the relationships between measurements and the causes of the faults in many instances can be obtained by reviewing the daily monitoring reports for a process, focussing particularly on those reports detailing when faults occurred and the investigation about the cause(s) of the faults. For this work, we picked examples for the five different faults cited above. Table 1 shows the relations between the values of the three measurements (normalized by dividing the measurement by the normal value of each process variable) and four different levels of deterioration for each of the five faults. The values listed for the three measurements and the normal values of variables were obtained by averaging for each measurement 1,000 samples generated from simulations representing normal or faulty conditions as the case would be, using the model in the Appendix. The fault data were generated by decreasing the parameters k_0, h, h', q, and q'. The number of the fault in Table 1 designates the label assigned to each of the five different faults, and the level of the fault corresponds to the degree of deterioration. Faults in level 1 are slight and thus incipient, faults in level 2 and 3 are medium, and a fault in level 4 is the most severe.

Table 1. Causes of Faults and Measurable Process Variables in the Quasisteady State*

Fault		Change in Variables Due to Faults		
No./Level	Cause	s_h^*	T_h^*	$C^*_{C_7H_8}$
1/1	0.90 k_o	0.98	0.99	0.95
2	0.85 k_o	0.97	0.99	0.92
3	0.80 k_o	0.96	0.99	0.89
4	0.75 k_o	0.95	0.99	0.86
2/1	0.90 h	1.07	1.02	1.00
2	0.85 h	1.12	1.03	1.00
3	0.80 h	1.17	1.04	1.00
4	0.75 h	1.22	1.06	1.00
3/1	0.90 h'	1.11	1.00	1.00
2	0.85 h'	1.18	1.00	1.00
3	0.80 h'	1.25	1.00	1.00
4	0.75 h'	1.33	1.00	1.00
4/1	0.90 q	0.95	0.99	1.05
2	0.85 q	0.92	0.98	1.07
3	0.80 q	0.90	0.97	1.11
4	0.75 q	0.87	0.97	1.14
5/1	0.90 q'	0.90	1.00	1.00
2	0.85 q'	0.85	1.00	1.00
3	0.80 q'	0.80	1.00	1.00
4	0.75 q'	0.75	1.00	1.00

*Values shown are normalized by the value of the variables in the normal steady-state condition: s_h^n = 223 mV, T_h^n = 889 K, $C^n_{C_7H_8}$ = 524 gmol/m^3, T_f^n = 300 K, $C^{in}_{C_7H_{16}}$ = 1,000 gmol/m^3

Training of the Neural Network to Diagnose Faults

Network architecture

Figure 3 shows the specific configuration of the network employed in this work. The network comprises three layers. It has three input nodes corresponding to the three measured variables, a middle layer with four nodes, and five output nodes corresponding to the five causes of faults. The number of nodes in the middle layer, namely four, was selected as optimal in minimizing the number of the training calculations. Table 2 shows the relation between number of nodes in the middle layer and number of training iterations to satisfy convergence condition $|t_{pj}^{(L)} - x_{pj}^{(L)}| < 0.1$ when a teaching pattern was fed to the network.

The parameters $\{\theta_j^{(1)}, j = 1, 2, 3\}$ in the sigmoid activation function (Eq. 1a) of three input nodes were fixed as follows:

$$\theta_j^{(1)} = 0, \quad j = 1, 2, 3$$

Input data (S_h, T_h, $C_{C_7H_8}$) to the network were normalized by

$$\overline{S_h} = 10 \cdot \frac{S_h - S_h^*}{S_h^*}, \quad \overline{T_h} = 10 \cdot \frac{T_h - T_h^*}{T_h^*},$$

$$\overline{C_{C_7H_8}} = 10 \cdot \frac{-C^*_{C_7H_8}}{C_{C_7H_8}}$$

so that the absolute value of $u_{pj}^{(\ell)}$, defined by Eq. 1b of the network, had a value of less than 2.2 (approximately) for any of the possible faults. Deviations of the process variables due to faults could thus be accurately caught by the network.

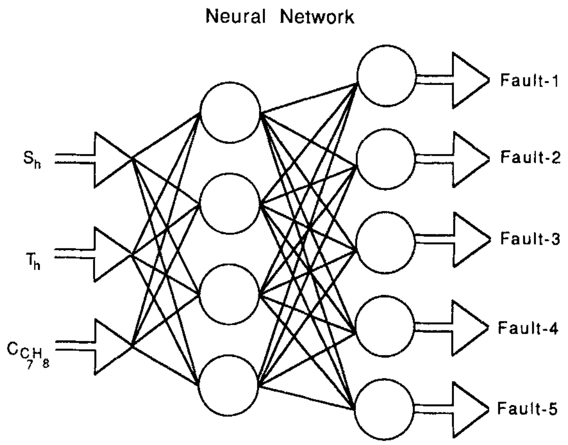

Figure 3. Three-layer artificial neural network for fault diagnosis of the example process.

Generation of teaching patterns

When the inputs in Table 1 were employed, we used the values for the measurements of the three process variables listed in Table 1, i.e., $s_h^* \cdot s_h^n$, $T_h^* \cdot T_h^n$, $C^*_{C_7H_8} \cdot C^n_{C_7H_8}$, as input, where the superscipt * designates the fractional change in the variable from its normal value. The outputs from the output nodes were trained to be composed of 0's and 1's. We trained the network to discriminate by using knowledge at level 1 of the five faults in Table 1.

Table 3 shows the training pattern used and the corresponding faults in level 1. The values of the three process variables used as the input teaching data were determined by averaging the simulated measurements over a period of 10 hours (1,000 samples) during which the fault occurred. Each output pattern corresponding to the faults was denoted by a vector. For example, the vector [1 0 0 0 0] was the output pattern for fault 1, and the vector [0 0 0 0 0] was the output pattern when no fault occurred.

Training

The network was trained using the backpropagation procedure. The iterative procedure [pattern p = 1 to N_p] was to introduce the training data set to the input nodes of the network

Table 2. Number of Nodes in the Middle Layer vs. Number of Iterations Engendered by Applying the Back-Propagation Training Procedure for the Network Architecture in Figure 2*

Nodes in Middle Layer	Iterations Used in Training with Quantitative Knowledge	
	Level 1	Level 3
2	>50,000	>50,000
3	>5,899	>3,122
4	>3,966	>2,118
5	>3,669	>1,753
6	>4,092	>2,033
7	>3,990	>2,013
8	>3,465	>1,714
9	>3,974	>2,026
10	>3,423	>1,639

*Teaching patterns are data of level 1 and level 3 in Table 1.

Table 3. Teaching Patterns for the Network*

Fault No./Level	Input Data Node 1 s_h	Input Data Node 2 T_h	Input Data Node 3 $C_{C_7H_8}$	Output Data *Node 1*	Output Data *Node 2*	Output Data *Node 3*	Output Data *Node 4*	Output Data *Node 5*
Fault 1/1	219.0	885.0	498.0	1	0	0	0	0
Fault 2/1	240.0	906.0	524.0	0	1	0	0	0
Fault 3/1	248.0	889.0	524.0	0	0	1	0	0
Fault 4/1	212.0	878.0	550.0	0	0	0	1	0
Fault 5/1	201.0	889.0	524.0	0	0	0	0	1
Normal	223.0	889.0	524.0	0	0	0	0	0

*Input data are the values of three process variables occuring under the faulty conditions at Level-1.

sequentially starting with the data of fault 1 and continuing to fault 5 followed by the normal data. Then the sequence would be repeated until criterion $|t_{pj}^{(L)} - x_{pj}^{(L)}| < 0.1$ was met. The learning parameters were chosen to be as follows: the learning rate constant $\eta = 0.1$, the momentum constant $\alpha = 0.9$, the initial thresholds $\theta_j^{(\ell)}$ were fixed values taken randomly from a uniform distribution in the range [0, 1]. The suboptimal weights were obtained by finding the set of $\{w_{ji}^{(\ell)}\}$ that minimized

$$E = \sum_{P=1}^{6} E_p \tag{3c}$$

with

$$E_P = \frac{1}{2}\sum_{j=1}^{5} \{t_{pj} - x_{pj}\}^2 \tag{3d}$$

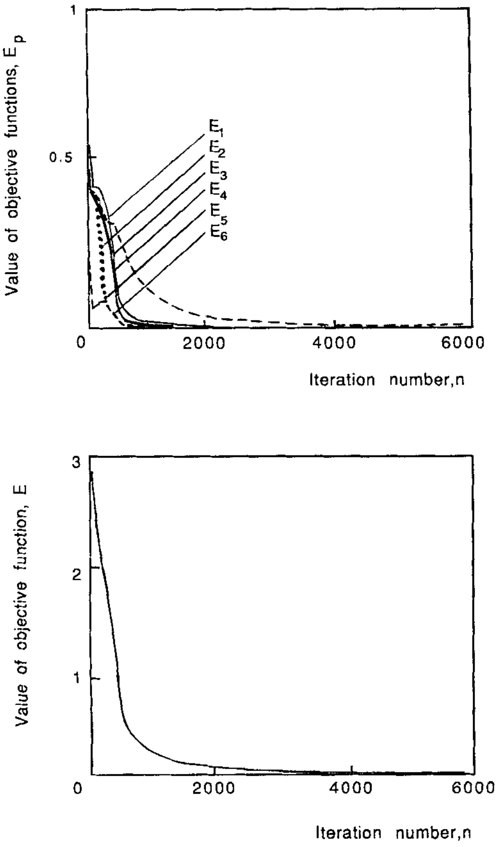

Figure 4. Convergence of the objective function.

from 1,000 sets of $\{w_{ji}^{(\ell)}\}$ in which each $\{w_{ji}^{(\ell)}\}$ was a random value that came from a uniform distribution with the range of $[-1,1]$.

Number of presentations for learning

The number of iterations of a pattern set required to learn the knowledge of the level 1 faults in Table 3 was 3,798.

The objective functions (E_1, E_2, E_3, E_4, E_5, E_6) and E decreased as the iterations increased (E_6 is the normal case). Figure 4 shows the decrease in the values of the objective functions.

Diagnosis of the Causes of Faults in the Example Process

Faults in the example process were diagnosed via the trained network. Initially, we simulated the operation of the reactor when the three process measurements were noise-free, and thereafter added noise to the measurements.

Table 4a. Average Activation Levels of Output Nodes of the Network*

Fault No./Level	Avg. Activation Level of Output Nodes Node 1	Node 2	Node 3	Node 4	Node 5
1/1	0.93**	0.06	0.00	0.00	0.02
2	0.99**	0.03	0.00	0.00	0.03
3	0.99**	0.01	0.00	0.00	0.05
4	1.00**	0.00	0.00	0.00	0.06
2/1	0.05	0.90**	0.03	0.00	0.00
2	0.12	0.92**	0.20	0.00	0.00
3	0.33	0.92**	0.49	0.00	0.00
4	0.73	0.95**	0.59	0.00	0.00
3/1	0.00	0.07	0.97**	0.02	0.00
2	0.00	0.00	1.00**	0.04	0.00
3	0.00	0.00	1.00**	0.06	0.00
4	0.00	0.00	1.00**	0.07	0.00
4/1	0.00	0.00	0.00	0.97**	0.02
2	0.00	0.00	0.00	1.00**	0.02
3	0.00	0.00	0.00	1.00**	0.02
4	0.00	0.00	0.00	1.00**	0.02
5/1	0.04	0.00	0.00	0.03	0.97**
2	0.06	0.00	0.00	0.05	0.99**
3	0.08	0.00	0.00	0.05	0.99**
4	0.09	0.00	0.00	0.05	1.00**

*The network was trained by knowledge of the faults in level 1 in Table 3. The three inputs to the network included noise.

**Percent diagnosed correctly.

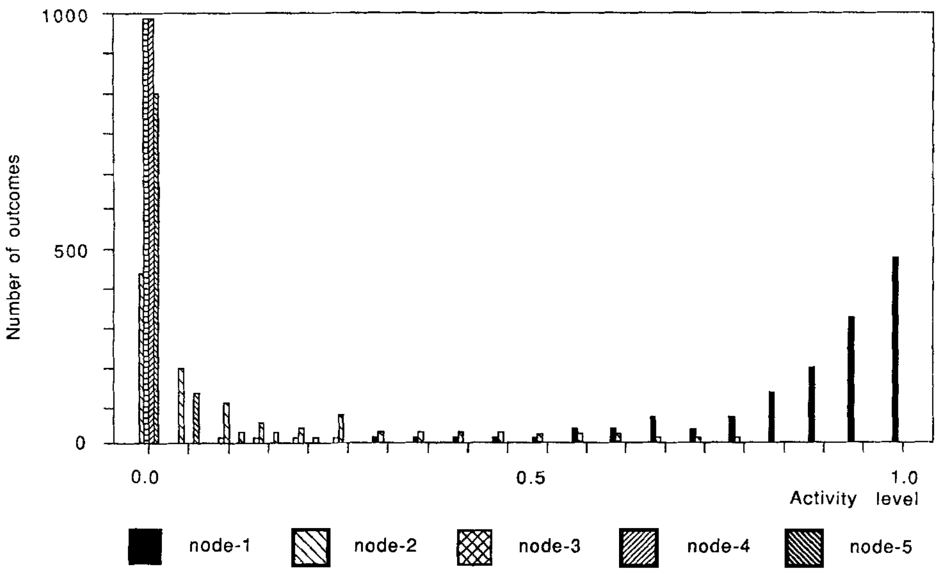

Figure 5. Activities of the output nodes of the network for 1,000 decisions (a total of 5,000 outputs) when fault 1 occurred.

The time interval for diagnosis was set to be 10 hours and the sampling interval for measurement was 0.01 hour so that 1,000 vectors corresponding to the three variables were introduced to the network for diagnosis. Every measurement was made under faulty conditions as well as for the normal condition for the process in the steady state.

We added measurement noise to the deterministic values of the three "measured" variables issuing from the simulation. The noise was normal random noise that had zero mean but had the following relative nonzero standard deviations:

- Outlet concentration of $C_{C_7H_8}$: 2.6% (5.2 gmol/m^3)
- Heater outlet temperature: 4.5% (4.5 K)
- Controller output signal: 7.3% (2.2 mV)

We carried out simulations of the reactor so that we could evaluate the ability of the network to discriminate among the causes of faults for a variety of levels of deterioration. For this work we used the network trained using the quantitative data in level 1 in Table 3. Table 4a shows the average values of 1,000 activity levels of the five output nodes of the network for each fault (fault 1 to fault 5) at each deterioration level (level 1 to level 4). For the case of fault 1, with any level of deterioration, when the measurements were fed to the network, node 1 always had the highest activity level among the five output nodes. Figure 5 shows the number of outcomes of the five nodes for each activity level when fault 1 occurred. Output node 1 had high activity levels whereas the other nodes had low activity levels. Similar results were obtained for the other respective faults. Thus, even though the network was trained by knowledge of faults at level 1, it could discriminate among the causes of faults for any level of deterioration in Table 1. Tables 4b and 4c show the classification matrices when faults at level 1 and level 3 occurred, respectively. These classification matrices demonstrate that the network correctly diagnosed the cause of the faults with reasonable robustness. Diagnostic results of faults at level 1 and level 2 via the neural network trained by the data at level 4 were not as satisfactory as these obtained via the network trained by the data at level 1.

Two-Stage Neural Network to Diagnose the Deterioration of Faults

One might train a multilayer neural network in such a way that the network could simultaneously diagnose the causes of faults and level of deterioration of a fault by using all of the quantitative knowledge in Table 1. Such a network would have three inputs and 20 [(the number of causes of faults) × (the number of levels of deterioration)] outputs. To train such a network would require quite extensive calculations. Besides, if you wanted to add one additional new level of deterioration or a new cause of a fault, the network would have to be trained all over

Table 4b. Classification Matrix for the Diagnosis Faults Caused by the Neural Network Trained Using the Knowledge of Faults in Level 1, Table 1*

Diagnosed / Introduced	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Normal
Fault 1	88.2%**	0.5%	0.0%	0.0%	0.0%	0.3%
Fault 2	2.0%	73.2%**	0.0%	0.0	0.0	0.0%
Fault 3	0.0%	0.3%	94.2%**	0.0%	0.0%	0.0%
Fault 4	0.0%	0.0%	0.0%	94.6%**	0.0%	0.0%
Fault 5	0.0%	0.0%	0.0%	0.0%	99.5%**	0.0%
Normal	0.4%	0.0%	0.0%	0.0%	0.0%	85.4%**

*The process measurements *included noise*. Faults introduced were of lowest level, such as the faults in level 1 in Table 1.
**Percent diagnosed correctly.

Table 4c. **Classification Matrix for the Diagnosis of Faults Caused by the Neural Network Trained Using the Knowledge of Faults in Level 3, Table 1***

Diagnosed / Introduced	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Normal
Fault 1	100.0%**	0.0%	0.0%	0.0%	0.0%	0.0%
Fault 2	14.8%	79.1%**	12.1%	0.0	0.0	0.0%
Fault 3	0.0%	0.0%	100.0%**	0.0%	0.0%	0.0%
Fault 4	0.0%	0.0%	0.0%	100.0%**	0.0%	0.0%
Fault 5	0.0%	0.0%	0.0%	0.0%	100.0%**	0.0%
Normal	0.2%	0.0%	0.0%	0.0%	0.0%	85.4%**

*The process measurements *included noise.* Faults introduced were of medium level, such as the faults in level 3 in Table 1.
**Percent diagnosed correctly.

again because the old knowledge stored in the network would not be adequate for discrimination.

Consequently, to alleviate such problems we present here an architecture for a two-stage neural network with each stage containing multilayers. The first stage is used to discriminate among the causes of faults, and the second stage is used to estimate the level of deterioration of a fault identified in the first stage. Figure 6 is a block diagram of the proposed two-stage network.

From the results described above, we know that a neural network trained using knowledge of the faults at deterioration level 1 discriminates among causes of faults even if other deterioration levels occur. Thus, the network in Figure 3 can be used as the first stage of the two-stage network.

After identifying the cause of the fault, the level of deterioration of the fault can be found from the second stage of the two-stage network. The same network configuration as shown in Figure 3 was used for the second stage. The same three process measurements were fed to the second stage of the network, and five nodes were the outputs representing five levels of deterioration of the fault. If more than five levels of deterioration are to be diagnosed, the number of the output nodes must, of course, be more than five.

We trained the five networks comprising the second stage using the data in Table 1. We also added data to make up a level between 0.9 and 1.0 for fault deterioration as listed in Table 5. Each respective network was used to diagnose the deterioration level for its associated fault. Table 6 shows the classification matrix based on 1,000 decisions for the diagnosis of the five faults. The respective matrices show that the five networks involved in the second stage of diagnosis clearly discriminated among the levels of deterioration of the faults. These results are encouraging because they demonstrate that use of a two-stage multilayer artificial neural network can diagnose incipient faults from three noisy process measurements.

A six-dimensional decision surface was mapped during the training phase. The decision surface for the example problem is complex, but can best be pictured as a six-dimensional pinball machine surface in which the holes (decisions) are not of uniform size and are surrounded by nonuniform slopes leading to the holes. A ball (a vector of inputs) located on an input surface is dropped and finds its way to the nearest hole. Each fault for each degree of error corresponds to a different hole, but the holes for one fault are distributed in associated regions representing each fault.

Thus, a ball that should be properly classified as fault x of

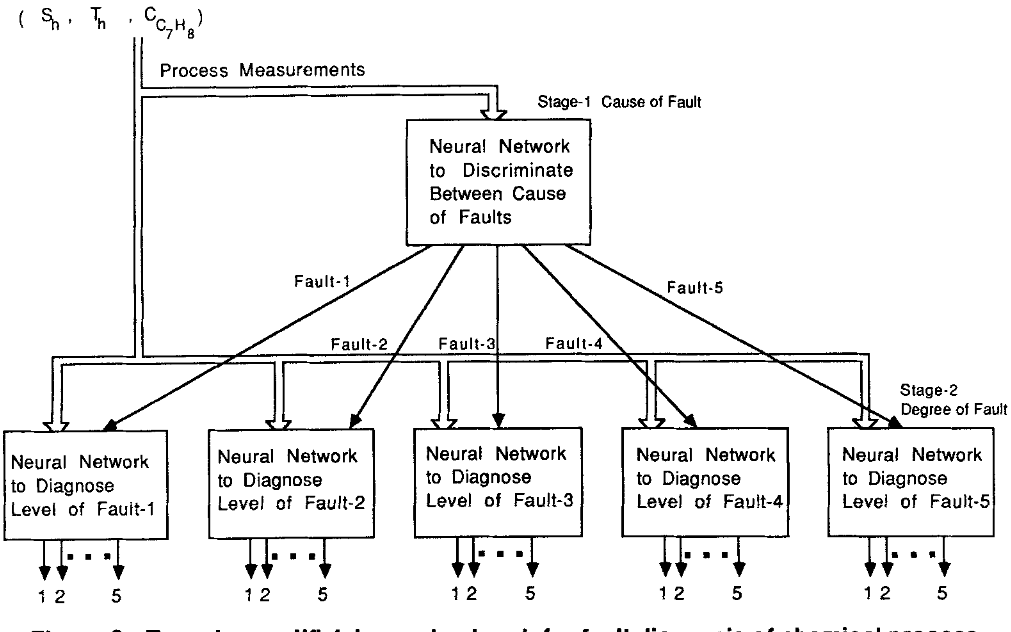

Figure 6. Two-stage artificial neural network for fault diagnosis of chemical process.
Each net has three layers.

Table 5. Causes of Very Slight Faults and Measurable Process Variables in the Quasisteady State*

Fault	Change in Variables Due to Faults			
No./Level	Cause	s_n^*	T_h^*	$C_{C_7H_8}^*$
1/½	$0.95k_o$	0.99	0.99	0.98
2/½	$0.95h$	1.04	1.01	1.00
3/½	$0.95h'$	1.05	1.00	1.00
4/½	$0.95q$	0.98	0.99	1.02
5/½	$0.95q'$	0.95	1.00	1.00

*Values shown are normalized by the value of the variable in the normal steady-state condition. This knowledge is used to diagnose the level of deterioration after the cause of faults is identified.

degree y might be misclassified as to the degree of fault x but is much less likely to be classified as a fault other than x.

The advantage of the two-stage network is that, if knowledge about the cause of a new fault becomes available, only the network in the first stage has to be retrained for diagnosis to occur. When new knowledge about the level of deterioration of a certain fault becomes known, only the one network corresponding to that fault has to be trained to accommodate the new knowledge.

Conclusions

In the two-stage network, the three-layer network in the first stage discriminated between the causes of faults, and the networks in the second stage identified the degree of the fault detected in the first stage. Even if the degree of fault differed, the network accurately discriminated among the faults in the first stage. The degree of the fault was accurately identified by the second stage.

We demonstrated how potential incipient faults in a chemical process, such as fouling of the heat exchanger surface, physical and/or chemical deterioration of the catalyst, partial plugging of the pipeline, and decrease in the heater performance, can be diagnosed via a proposed two-stage artificial neural network with each stage composed of three layers.

Although we used simulations to generate data that were used in training the network to discriminate, and flow sheeting codes could be employed for the same purpose, it is also possible to collect data about faults and related measurements from the historical databases in the plant and use the historical information to train the network. We used noisy measurements of the three process variables for training, and plant data would be even noisier and have occasional gross errors in the data. Artificial neural networks, however, exhibit the ability to filter out the noise and intermittent biases.

We conclude that artificial neural network have considerable potential in fault detection and trouble-shooting in chemical plants.

Notation

a = area of heat exchange
a' = area of heat exchange of heater
$C_{C_7H_{16}}$ = concentration of heptane
$C^i_{C_7H_{16}}$ = inlet concentration of $C_{C_7H_{16}}$
$C_{C_7H_8}$ = concentration of toluene
$C^n_{C_7H_8}$ = normal concentration of toluene

Table 6. Classification Matrix for the Diagnosis of the Level of Deterioration*

Introduced		Diagnosed				
		Level 1/2	Level 1	Level 2	Level 3	Level 4
Fault 1	Level 1/2	83.7%**	4.2%	0.0%	0.0%	0.0%
	Level 1	8.6%	62.7%	13.3%	0.0%	0.0%
	Level 2	0.0%	12.4%	75.7%**	12.5%	0.0%
	Level 3	0.0%	0.0%	3.2%	72.6%	2.8%
	Level 4	0.0%	0.0%	0.0%	1.5%	85.3%**
Fault 2	Level 1/2	82.9**	2.8%	0.0%	0.0%	0.0%
	Level 1	7.5%	62.9%**	16.5%	0.0%	0.0%
	Level 2	0.0%	21.9%	62.7%**	9.8%	0.5%
	Level 3	0.0%	0.4%	8.9%	48.5%**	17.9%
	Level 4	0.0%	0.0%	0.0%	15.7%	70.1%**
Fault 3	Level 1/2	97.6%**	0.1%	0.0%	0.0%	0.0%
	Level 1	1.9%	76.9%**	18.4%	0.0%	0.0%
	Level 2	0.0%	21.6%	66.9%**	11.4%	0.0%
	Level 3	0.0%	0.0%	1.2%	58.7%**	14.9%
	Level 4	0.0%	0.0%	0.0%	12.9%	69.5%**
Fault 4	Level 1/2	94.6%**	0.3%	0.0%	0.0%	0.0%
	Level 1	1.7%	73.9%**	7.8%	0.0%	0.0%
	Level 2	0.0%	10.2%	80.6%**	2.8%	0.0%
	Level 3	0.0%	0.0%	0.0%	86.0%**	0.4%
	Level 4	0.0%	0.0%	0.0%	0.0%	92.1%**
Fault 5	Level 1/2	90.6%**	0.4%	0.0%	0.0%	0.0%
	Level 1	0.0%	98.1%**	0.0%	0.0%	0.0%
	Level 2	0.0%	0.8%	80.1%**	7.1%	0.0%
	Level 3	0.0%	0.0%	1.8%	81.9%**	0.4%
	Level 4	0.0%	0.0%	0.0%	0.0%	99.9%**

*The network was trained using the knowledge listed in Tables 1 and 5. The process measurements included noise.
**Percent diagnosed correctly.

$C^*_{C_7H_8}$ = concentration of toluene normalized by its normal value
C_{H_2} = concentration of H_2
C_p = specific heat
C'_p = specific heat
c = recycle rate
cT = inlet temperature of heating water
E_a = activation energy
ΔE = heat of reaction
h = overall heat transfer coefficient of reactor
h' = overall heat transfer coefficient of heater
K = heater gain
$K_{mV/T}$ = gain of temperature mV transducer
k = rate constant of reaction
k_c = proportional gain of controller
k'_h = overall gain from heater driving signal to heater output temperature
k_o = frequency factor of catalyst
L = number of total layers of the multilayer network
ℓ = number of the layer of a multilayer neural network
N = total number of layers of a multilayer neural network
$N^{(\ell)}$ = number of nodes in the ℓ layer of a multilayer network
n = discrete time in learning
p = pattern identification number
q = volumetric flow rate of heptane
q' = volumetric flow rate of steam
R = gas constant
s_h = controller output signal
s^n_h = normal controller output signal
s^*_n = controller output signal normalized by its normal value
s_i = output of integrator in PI controller
T = reaction temperature
T_h = heater outlet temperature
T^n_h = normal heater outlet temperature
T^*_h = heater outlet temperature normalized by its normal value
T_i = temperature of reactor inlet stream
T^*_i = integration time of controller
$T^{(\ell)}_j$ = parameter to adjust characteristic of activation of node j
$t^{(\ell)}_{pj}$ = teaching signal of output of node j in the ℓ layer
u_c = command signal
$u^{(\ell)}_{pj}$ = summation of input to node j in the ℓ layer when pattern p is fed
V = effective reactor volume
V' = effective volume of the heater
$w^{(\ell)}_{ji}$ = weight from ith node i in $(\ell - 1)$ layer to jth node in ℓ layer
$\Delta w^{(\ell)}_{ji}$ = adjustment of $w^{(\ell)}_{ji}$
$x^{(\ell)}_{pj}$ = output of node j in the ℓ layer when an input pattern p is fed

Greek letters

α = momentum term to smooth weight changes
$\delta^{(\ell)}_{pj}$ = leaning signal
η = learning rate constant
$\theta^{(\ell)}_j$ = threshold of node j in the ℓ layer
ρ = density
τ = heater time constant

Overlay

$\cdot$ = derivative

Literature Cited

Aso, H., *Information Processing via Neural Networks*, in Japanese Sangyo Publishing Co., Tokyo, Japan (1988).

Beard, R. D., "Failure Accommodation in Linear Systems Through Self-Recognition," Rept.MVT-71-7, Man-Vehicle Laboratory, Cambridge, MA (1971).

Dietz, W, E., E. L. Kiech, and M. Ali, "Classification of Data Patterns Using an Autoassociative Neural Network Topology," Int. Conf. Indus. Engr. Applns. of AI & Expert Systems, Tullahoma, TN (June 6–9, 1989).

Gallant, S., "Automated generation of connectionist expert systems for problems involving noise and redundancy," AAAI Workshop on Uncertainty (1987).

Himmelblau, D. M., "Fault Detection and Diagnosis in Chemical and Petrochemical Processes," *Chem. Eng. Monographs*, No. 8, Elsevier (1978).

Hopfield, J. J., "Neural Network and Physical Systema with Emergent Collective Computational Abilities," *Proc. Nat. Sci. U.S.A.*, *72* (Apr., 1982).

Hopfield, J. J., and D. W. Tank, "Computing with Neural Circuits: A Model," *Sci.*, **233**, 4764 (1986).

Hoskins, J. C., and D. M. Himmelblau, "Neural Network Models of Knowledge Representation in Process Engineering," *Comp. in Chem. Eng.*, **12**, 881 (1988).

Kramer, M. A., "Malfunction Diagnosis Using Quantitative Models with Non-Boolean Reasoning in Expert Systems," *AIChE J.*, **33**, 130 (1987).

———, "Narrowing Diagnostic Focus Using Functional Decomposition," *AIChE J.*, **34**, 25 (1988).

Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.* (Apr. 4, 1987).

McClelland, J. L., and D. E. Rumelhart, eds., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," Vols. 1, 2, 3, MIT Press, Cambridge, MA (1986).

Naidu, S., E. Zafiriou, and T. J. McAvoy, "Application of Neural Networks on the Detection of Sensor Failure During the Operation of a Control System," ACC Meeting, Pittsburgh (1989).

Ungar, L. H., and B. Powell, "Fault Diagnosis Using Connectionist Adaptive Networks," AIChE meeting, Washington, DC (Nov. 27–Dec. 2, 1988).

Watanabe, K., "Introduction to Fault Diagnosis of Industrial Systems," *Daily Ind. Newspaper*, Tokyo (1983).

Watanabe, K. and D. M. Himmelblau, "Fault Diagnosis in Nonlinear Chemical Processes: I. Theory," *AIChE J.*, **29**, 243 (1983a).

———, "Fault Diagnosis in Nonlinear Chemical Processes: II. Application to a Chemical Reactor," *AIChE J.*, **29**, 250 (1983b).

———, "Incipient Fault Diagnosis of Nonlinear Processes with Multiple Causes of Faults," *Chem. Eng. Sci.*, **39**, 491 (1983).

Willsky, A. S., "A survey of Design Methods for Failure Detection in Dynamic Systems," *Automat.* **12**, 601 (1976).

Appendix: Process Model

The lumped model of the reactor, heater, and controller in Figure 2 is:

Reactor energy balance:

$$k(T) = k_o \exp(-E_a/RT) \quad \dot{T} = \frac{q}{V}(T_i - T) - \frac{\Delta H}{\rho C_p} k(T) C_{C_7H_{16}} + \frac{ah}{\rho C p V}(T_h - T), \qquad T(0) = T_o$$

We have assumed the specific heat C_p and density ρ are the same for the inlet and outlet streams to keep the presentation simple.

Reactor mass balance:

$$\dot{C}_{C_7H_8} = -\frac{q}{V} C_{C_7H_8} + k(T) C_{C_7H_{16}}, \quad C_{C_7H_8}(0) = C^0_{C_7H_8}$$

$$\dot{C}_{H_2} = -\frac{q}{V} C_{H_2} + 4k(T) C_{C_7H_{16}}, \quad C_{H_2}(0) = C^0_{H_2}$$

$$\dot{C}_{C_7H_{16}} = -\frac{q}{V} C_{C_7H_{16}} - k(T) C_{C_7H_{16}} + \frac{q}{V} C_{C_7H_{16}}, \quad C_{C_7H_{16}}(0) = C^0_{C_7H_{16}}$$

Not all above equations are independent.

Heater energy balance:

$$\dot{T}_h = \frac{1}{\tau}(cT - T_h) + \frac{K}{\tau} s_h, \qquad T_h(0) = T_{ho}$$

PI controller:

$$\dot{s}_i = \frac{K_c}{T_i^*}(u_c - K_{mV/T} \cdot T), \quad s_i(0) = s_{io}$$

$$s_h = k_c(u_c - K_{mV/T} \cdot T) + s_i$$

The process variables at the steady-state operating point and the process coefficients had the following values for the simulations:

State variables:

T = reaction temperature, 740.0 K [Range: 673 ~ 773 K]
$C_{C_7H_8}$ = outlet concentration of C_7H_8, 524.0 gmol/m³ [Range: 400 ~ 600 gmol/m³]
C_{H_2} = utlet concentration of H_2, 2,097.0 gmol/m³
$C_{C_7H_{16}}$ = outlet concentration of C_7H_{16}, 476.0 gmol/m³
T_h = heater outlet temperature, 889.0 K
s_i = output of integrator in PI controller, 223.0 mV [Range: 202 ~ 232 mV]

Intermediate variable:

s_h = driving signal for heater (output of the PI controller), 223.0 mV

Process inputs:

T_i = temperature of reaction inlet stream, 300.0 K
$C^i_{C_7H_{16}}$ = inlet concentration of C_7H_{16}, 1,000.0 gmol/m³
cT = inlet temperature of heating water, 0.9 T K
u_c = command signal, 740.0 mV

Reactor:

k_o = frequency factor, 5.01×10^8 1/h
E_a = activation energy, 1.369×10^5, J/gmol
R = gas constant, 8.319 J/gmol · K
ΔH = heat of reaction
$= 2.2026 \times 10^5 + 6.2044 \times 10^1 T - 5.536 \times 10^{-2} T^2 - 1.15 \times 10^{-6} T^3 + 3.1496 \times 10^{-7} T^4$, J/gmol
C_P = specific heat, 490.7 J/gmol · K
ρ = density, 593.0 gmol/m³
a = area of heat exchange, 10.0 m²
h = overall heat transfer coefficient, 6.05×10^5 J/m² · h · K
q = inlet and outlet volumetric flow rate, 3.0 m³/h
V = effective reactor volume, 30.0 m³

Heater:

$\tau = V'/q'$ = heater time constant, 0.2 h
$K = a'h'k'_h/(\rho' C'_P q')$ = heater gain, 1.0 K/mV

PI controler:

k_c = proportional gain, 20.0
T_i^* = integrator coefficient, 0.3 h
$K_{mv/T}$ = gain of temperature to mV transducer, 1.0 mV/K

Manuscript received Feb. 14, 1989, and revision received July 31, 1989.