Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks

J. C. Hoskins, K. M. Kaliyur, and David M. Himmelblau
Dept. of Chemical Engineering, University of Texas, Austin, TX 78712

In some research areas, investigators use artificial neural networks in an attempt to mimic certain processes found in biological and human nervous systems. It would be a welcome event if artificial neural networks could capture the productive, creative and holistic aspects of human intelligence that underlie people's ability to spontaneously learn new things and instantly perform such tasks as pattern recognition and association in real time, even when faced with incomplete or distorted input. Such, however, is not yet the case. Consequently, it is better, at least at this time, to view an artificial network used for fault detection and diagnosis as an adjustable modeling device that can transform current data into a readily discernable form of information in real time.

Fault detection and diagnosis essentially are tasks of pattern recognition. Sensor (and other) data, which contain no readily discernable message, can be transformed via pattern recognition into clear cut information useful for decision-making. Since artificial neural networks (after training) classify data effectively, it would seem that an artificial neural network would be an appropriate tool to try for fault diagnosis in complex plants.

Artificial neural networks and traditional pattern classifiers were compared in several studies (Burr, 1988; Huang and Lippman, 1987; Lucas and Kittler, 1989), in which their relative performance was evaluated. Such studies show that artificial neural networks with unsupervised learning can be the equivalent of *k*-nearest neighbor or *k-nn* classifiers [*k-nn* classifiers are known to produce decision boundaries that approximate the optimal Bayes decision rule classification (Devijver and Kittler, 1982)]. Experiments comparing one artificial neural network, namely Kohonen's self-organizing feature map learning algorithm (Kohonen, 1984), with the MULTIEDIT/CONDENSING algorithm (Voisin and Devijver, 1987) and with the *k-nn* algorithm indicated little difference in their effectiveness in classifying speaker-independent speech.

The purpose of our investigation is to illustrate a neural network approach to fault detection and diagnosis in a large complex chemical plant. We demonstrate the ability of a neural network to learn nonlinear mappings in the presence of noisy inputs. We also show that neural network models can exhibit the rule-following behavior of knowledge-based expert systems without containing any explicit representations of the rules. We conclude that, given a valid database, fault detection and diagnosis is a promising area for the application of artificial neural networks in industry.

Fault Detection and Diagnosis

We are interested in a process that is characterized by hundreds of state variables, not all of which may be measured (probably sampled periodically). A control system exists, and certain controlled variables have specifications (constraints) that must be met. From the view point of the operator who supervises the time performance of the equipment, the process is operating *normally* if its observed and particularly its controlled variables are in the neighborhood of their desired values. A *fault* (or state of fault) occurs when a certain level of deterioration takes place in one or more of the states because of temporary or permanent physical changes (scaling, tube plugging, sensor deterioration, leaks, etc.). Failure is the complete degradation of performance. Faults may not only just lead to poor economy but also lead to catastrophic events, in which equipment is damaged or people get hurt.

Ordinarily, certain variables are observed by either a human or a computer. The decision as to what to do about a fault—nothing, wait until the next scheduled maintenance to correct, shutdown immediately, and so on—is usually made by a human who would like to:

1. Learn about the existence and degree of a fault as soon as possible (incipient fault detection)
2. Isolate the type of fault (fault localization)
3. Identify the physical cause(s) of the fault.

On-line fault detection and diagnosis are particularly desirable. Incipient fault detection, i.e., detection at the earliest possible stage, is the desired goal. In the face of the increasing complexity and automation of plants, achieving this objective requires effective, economical techniques.

Various quantitative methods to diagnose incipient faults can be found in the literature (Basseville and Benveniste, 1986; Collacott, 1977; Himmelblau, 1978; Isermann, 1984; Lieberman, 1985; Lyon, 1987; Pan, 1981; Patton et al., 1989; Sacks and Liberty, 1974). For example, Beard (1971) presented a filtering (full-order Luenberger observer) technique that generated estimation error 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 a nonlinear estimator to identify process parameters indicative of process faults caused by deterioration of components. Fault detection via a state space approach is difficult because the technique assumes that the process model is known quite well. In practice, it is impossible to exactly pinpoint what is going on, hence fault detection and diagnosis can be quite difficult because errors in the model can be interpreted as faults, thus yielding false alarms, or prevent faults from being detected when they occur.

Rule-base expert systems with Boolean (binary) reasoning or those with non-Boolean (for example, fuzzy) reasoning are one way to represent knowledge about faults; they can be effectively applied for 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 functional decomposition (Kramer, 1988). Fault diagnosis via rule-based expert systems, however, does require a base 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 malfunction. How accurate a diagnosis is depends on how rich and accurate the rule base is, and forming the rule base itself is quite time-consuming (and expensive) and is perhaps as difficult a job as process modeling.

Artificial Neural Networks

Artificial neural networks recently have been proposed as tools for fault detection and diagnosis (Dietz et al., 1989; Hoskins and Himmelblau, 1988; Naidu et al., 1989; Ungar et al., 1989; Venkatasubramanian and Chen, 1989; Venkatasubramanian et al., 1990). In applying artificial neural networks to fault detection, we would like for the network to perform several functions (that might be carried out by separate means in other techniques of fault detection):

1. Classification of labeled input data used in training to provide a clear cut labeling of a fault, i.e., partitioning of input patterns into a prespecified number of groups or classes and correct classification of new data not immediately discernible to a human being
2. Formation of associative memory, i.e., identification of a fault from partial or corrupted input
3. Nonlinear mapping of a vector of inputs into a vector of outputs (via training)
4. Generalization, that is, (for fault detection) appropriate classification in the face of noise and distortion of inputs.

Our work is focused on functions 1 and 3, and noise in 4.

One of the advantages of artificial neural networks is that sometimes classification can be accomplished more rapidly than classification using statistical packages (such as SAS or SPSS). Of course, the disadvantage is that training via a back-propagation algorithm is much slower than training required by a statistical package.

Both MULTIEDIT/CONDENSING and artificial neural networks aim to build a small reference set of vectors to be used in the classification. MULTIEDIT/CONDENSING edits a large number of elements from the training set—perhaps as

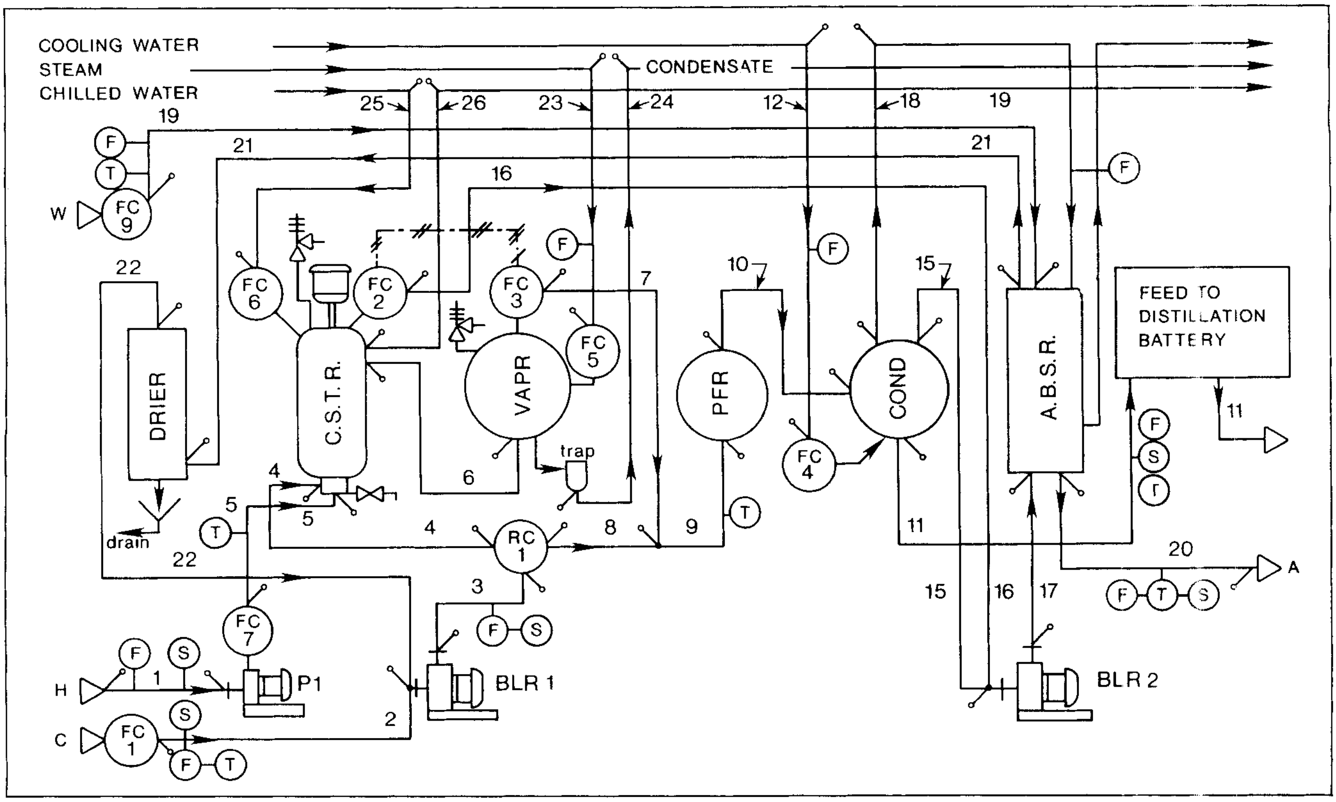

Figure 1. Syschem plant.

many as 95% of the elements. Kohonen's algorithm builds up a small set of reference vectors. In our work, we used a three-layer artificial neural network with two "hidden" layers and backpropagation so that the transformation between the input layer and the first hidden layer was carried out the usual (nonlinear) principal component analysis (Oja, 1989). The number of hidden layers and the number of nodes in each layer were determined by computer experiments designed to minimize computation time and yet achieve suitable classification. With 418 sensor measurements (and thus 418 input nodes in the network), and 40 nodes in the first hidden layer, the net we used effectively implemented about a 90% reduction in the input vector elements. The principal components might well be regarded as features fed to the second hidden layer (comprising from 18 up to 30 nodes), which in turn fed their output layer which contained 20 nodes representing 19 faults and the normal state.

The Syschem Plant

To test the capability of a neural network to identify and isolate faults, we used a chemical plant simulator, the Syschem plant (Doig, 1986). The Syschem plant consists of various units such as a drier, liquid-phase CSTR reactor, vaporizer, adiabatic plug flow reactor, condenser, absorber column, and various control valves and pumps, see Figure 1. In the liquid-phase isothermal CSTR, liquid hydrocarbon reacts with gaseous chlorine to produce liquid monochloride and hydrogen chloride gas. Cooling water maintains the desired temperature. The vaporizer vaporizes the liquid product from the CSTR. Subsequently in the adiabatic plug flow reactor (PFR), additional reaction of monochloride takes place to produce dichloride. The condenser condenses the hydrocarbon and other components in the stream leaving the PFR. The absorber removes HCl from the gaseous products of the CSTR and PFR reactors by scrubbing the gas streams with water to make 30 wt. % commercial-grade hydrochloric acid. The system absorbs HCl at relatively high temperatures within the column. The total number of available sensor measurements in the plant was 418, representing 11 properties of 38 streams. A fault was to be diagnosed on the basis of these measurements. All of the 418 measurements for each fault condition were scaled suitably by taking the maximum deviation of a value of a variable from its normal value (+ or −), and making the absolute value of that deviation correspond to +1, and −1, respectively, for the input to the node. Although we realized that having 418 measurements would force the network to deal with a significant number of interconnections and thus require more calculations and weight adjustments during each iteration of the training phase, we hoped that the detailed input data would provide better classification of the faults and that the network also could easily be modified to include the analysis for more faults if required later on.

Nineteen different fault states plus the normal state were introduced into the Syschem plant simulator. Table 1 lists the faults. We added normal random noise with coefficients of variation up to 0.10 to each of the input vectors associated with the 19 faults and the normal state to generate 200 patterns for training and 200 patterns for testing for each level of noise. The introduction of noise was intended to represent uncertainty in sensor measurements that would exist in practice. Consequently, the 20 output nodes also had random outputs, but one output could easily be distinguished as being close to 1 (in the range of 0.80 to 0.95) and all the others would be close to zero (in the range of 0 to 0.15).

Table 1. Fault States Employed in the Syschem Plant

No.	Condition
1	Excessive heat loss from vaporizer whose insulation may be damaged or wetted
2	Cooling coil in the CSTR fouled and heat transfer resistance too high
3	Condenser partially flooding
4	PFR not operating adiabatically; cooling occurring and insulation possibly damaged
5	Process in the condenser gaining heat; an unwanted chemical reaction possibly occurring within the condenser shell
6	Reactants escaping from the PFR due to a leaking flanged head and joint
7	Level of the reactants in the CSTR too high
8	Gas entering into vaporizer feed resulting in further reaction
10	Control valve controlling steam 8 not fully open
11	Loss of catalyst activity in PFR
12	"Hot Spot" formed in the middle catalyst bed in the PFR resulting in very high conversion
13	Gross imbalance in residence times of reactants in the catalyst bed caused by channeling of gas through the bed in PFR
14	Resistance to the liquor flow through interstage cooler 1 in the absorber becoming excessive resulting in a drop in performance
15	Excessive liquor level in the vaporizer possibly caused by a faulty controller system (FC5)
16	Desiccant in the drier saturated resulting in a need to regenerate the drier
17	Resistance to gas flow through stage 1 tray in the absorber becoming excessive resulting in low efficiency
18	Desiccant quality becoming poor or massive corrosion occurring in the drier; unexpected reaction occurring with the process gas
19	Excessive resistance to the cooling water flow through interstage cooler 2 in the absorber

Results

Because of the size of the network, only some of the more interesting results can be presented here. In general our results agreed with those of other investigators for the selected type of network, namely the number of presentations required to reach a given state of learning (a certain precision for the activities of the elements of the output vector) depended on the number of nodes in the hidden layer and asymptotically reached a minimum. Two hidden layers had to be used due to the complex nature of the plant. After some preliminary trials, the number of nodes in the first hidden layer was kept constant at 40, while the number of nodes in the second hidden layer was varied from 18 to 30. Examine Table 2. More than 30

Table 2. Test Results: Number of Iterations in the Learning Phase As a Function of Number of Nodes in the Second Hidden Layer*

No. of Nodes in Hidden Layer 1	No. of Nodes in Hidden Layer 2	No. of Iterations to Reach Error Tolerance of 0.05
40	18	17,600
40	20	14,000
40	30	10,800

*Learning rate = 0.06; momentum = 0.9; coefficient of variation of noise in input vector elements = 0.05

nodes in the second hidden layer increased the number of required iterations. Although the computation time is not particularly meaningful because it depends so much on the hardware and software employed, the net with 418/40/30/20 nodes took 24 hours on a Sun 4/410 workstation to converge to the specified tolerance of 0.05. (Each of the deviations between the target values for the output nodes and the propagated values for any input pattern was ≤ 0.05.) The literature has many suggestions as to how to speed up the learning phase.

Probably the factor of most interest is the degree of success of the network in properly classifying vectors of input measurements that were not used in the training phase, particularly as a function of noise in the sensor data. Because we used a simulator, we could control the extent of the noise. For the low-level, normally-distributed noise (a coefficient of variation was 0.01), the classification of new vectors not used in the training was perfect without the need for statistical tests. For higher-level noise (a coefficient of variation of 0.10), more presentations for training were required, but the classification was still perfect. These results illustrate the ability of an artificial neural network to filter out noise. Table 3 shows the values of the output nodes for a typical input vector not used in the learning phase that included a medium noise level (a coefficient of variation of 0.05) in the input vector elements. One can easily classify the fault (correctly) without statistical discrimination tests as shown by the main diagonal of Table 3. Only for fault 11 could there be any possible question about the classification. It took only 3 or 4 seconds to produce the data in Table 3 once the network was trained. If a vector to be classified is related to a fault not present during the training phase, then the values of the output nodes will fall into two categories. One category is to misclassify the vector as being similar to one of the faults used in the training phase. Even though we did not study this network function, we believe that this outcome can usually be detected because the ratio of the value of the output node having the highest value to the one with the next highest values is near 1 in contrast to the much larger ratio observed for the cases listed in Table 3. For example, for the normal state in Table 3, the ratio is 0.916/0.358 = 2.56 and for fault 1 the ratio is 0.908/0.112 = 8.11. For fault 11 the ratio is 0.672/0.506 = 1.33, hence there might be some question as to whether the vector associated with fault 11 was indeed fault 11 or another fault. The other category that might occur for a new fault is not so much misclassification as confusion as to what the classification should be. Several output nodes would have relatively the same values so that several ratios would be close to each other and near 1.

Implementation in a Plant Environment

How can artificial neural networks be implemented in a plant if an engineer decides that they might be a useful tool for fault

Table 3. Classification of Faults for Sensor Measurements with a Medium Noise Level† for an Input Vector not Used in Training (Output Node Activities are Listed)

Actual Fault	0*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0.916	0.098	0.096	0.097	0.104	0.100	0.099	0.099	0.100	0.096	0.093	0.104	0.102	0.099	0.101	0.103	0.100	0.099	0.077	0.100
1	0.119	0.908	0.084	0.101	0.098	0.101	0.101	0.099	0.100	0.100	0.100	0.101	0.100	0.101	0.101	0.100	0.104	0.100	0.100	0.100
2	0.097	0.092	0.886	0.100	0.102	0.099	0.100	0.099	0.099	0.097	0.099	0.098	0.099	0.099	0.098	0.098	0.100	0.099	0.100	0.099
3	0.000	0.112	0.000	0.871	0.000	0.001	0.093	0.000	0.010	0.088	0.144	0.002	0.007	0.008	0.075	0.102	0.000	0.017	0.000	0.088
4	0.086	0.102	0.113	0.099	0.899	0.099	0.099	0.100	0.100	0.101	0.101	0.100	0.098	0.100	0.100	0.100	0.097	0.099	0.097	0.099
5	0.110	0.096	0.096	0.095	0.113	0.895	0.110	0.094	0.103	0.091	0.081	0.098	0.103	0.093	0.103	0.100	0.060	0.096	0.049	0.102
6	0.016	0.101	0.117	0.094	0.090	0.107	0.883	0.100	0.099	0.109	0.109	0.104	0.096	0.109	0.101	0.100	0.098	0.100	0.118	0.099
7	0.128	0.096	0.096	0.025	0.036	0.102	0.095	0.856	0.099	0.007	0.139	0.016	0.104	0.064	0.098	0.096	0.087	0.069	0.115	0.101
8	0.153	0.095	0.097	0.102	0.061	0.092	0.097	0.102	0.890	0.102	0.116	0.092	0.092	0.097	0.091	0.095	0.100	0.100	0.100	0.093
9	0.000	0.083	0.099	0.191	0.088	0.091	0.105	0.003	0.084	0.822	0.120	0.506	0.065	0.128	0.025	0.072	0.096	0.040	0.132	0.063
10	0.000	0.062	0.076	0.101	0.000	0.083	0.086	0.099	0.095	0.071	0.857	0.004	0.108	0.108	0.084	0.090	0.029	0.098	0.104	0.095
11	0.358	0.000	0.092	0.004	0.032	0.007	0.001	0.129	0.096	0.149	0.012	0.672	0.063	0.058	0.000	0.099	0.085	0.119	0.089	0.001
12	0.124	0.085	0.091	0.117	0.064	0.085	0.097	0.105	0.093	0.071	0.129	0.052	0.879	0.104	0.086	0.098	0.088	0.085	0.104	0.093
13	0.084	0.090	0.100	0.101	0.107	0.103	0.101	0.100	0.100	0.099	0.099	0.102	0.101	0.897	0.101	0.101	0.113	0.101	0.101	0.100
14	0.112	0.097	0.095	0.100	0.076	0.096	0.100	0.097	0.099	0.095	0.098	0.097	0.099	0.095	0.889	0.098	0.101	0.098	0.095	0.100
15	0.266	0.105	0.105	0.099	0.090	0.098	0.100	0.099	0.100	0.099	0.095	0.098	0.101	0.097	0.099	0.899	0.100	0.100	0.103	0.100
16	0.086	0.085	0.000	0.098	0.130	0.100	0.093	0.104	0.097	0.102	0.105	0.095	0.097	0.108	0.096	0.098	0.875	0.100	0.095	0.097
17	0.038	0.107	0.106	0.062	0.219	0.100	0.097	0.085	0.101	0.026	0.125	0.152	0.098	0.086	0.115	0.103	0.072	0.878	0.101	0.102
18	0.096	0.103	0.099	0.100	0.100	0.101	0.101	0.101	0.100	0.102	0.101	0.100	0.101	0.100	0.101	0.101	0.100	0.101	0.904	0.100
19	0.106	0.104	0.104	0.098	0.105	0.102	0.098	0.101	0.100	0.101	0.096	0.100	0.102	0.101	0.101	0.102	0.103	0.101	0.100	0.900

*Normal case
†Coefficient of variation = 0.05

diagnosis? Under existing conditions, it is unlikely that data would have been collected to be used to train a proposed network. Two considerations are suggested here:

1. Plan and execute a program analogous to that for statistical quality control by which input sensor measurements and other data (including qualitative variables) are expressly related to faults and their causes. If a program for quality control exists, this suggestion should not be too costly. A network would have to be continuously updated as the plant equipment, feeds, and environment change so that a dual network might be used: one always in training and one in operation, with the latter periodically being replaced by the updated network.

2. Develop (or modify an existing) plant simulator so that it represents the sections of the plant of interest reasonably well, but not necessarily perfectly. It is easy to introduce faults into the simulator as we have done and train a network for fault diagnosis. The results of simulation can be compared with whatever valid historical data exist, and the network "tuned" with the data. Use of a simulator plus the plan described above in 1 would be the best arrangement.

Conclusions

Artificial neural networks exhibit a number of features that make them attractive for fault detection and diagnosis in complex systems. A network can learn the correct associations between system faults and vectors of sensor measurements. Furthermore, a network can generalize so that input patterns not in the training set can be classified. Finally, artificial neural networks can accommodate their diagnosis to the noise and uncertainty that exist in all process measurements.

Three factors need to be mentioned in connection with the potential practical implementation of the technique we have used. First, the Syschem plant simulator represented a steady-state plant. If distinct trends, rather than fluctuations, occur in plant data, the training of a network must involve time-dependent data, and hence a much bigger network must be used. Second, training for fault detection and diagnosis requires a database that may not exist. If historical data are all that are available, a special effort would have to be carried out to relate faults to sensor measurements. Finally, high-speed parallel hardware is really essential if thousands of sensor measurements are to be used in training a network. In spite of these caveats, artificial neural network to us appear to be a promising tool to supplement decision-making in real time.

Literature Cited

Basseville, M., and A. Benveniste, eds., "Detection of Abrupt Changes in Signals and Dynamic Systems," *Lecture Notes in Control and Sciences*, **77**, Springer Verlag, New York (1986).

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

Burr, D. J., "Experiments on Neural Net Recognition of Spoken and Written Text," *IEEE Trans. ASSP*, **36**, 1162 (1988).

Collacot, R. A., *Mechanical Fault Diagnosis*, Chapman and Hall, London (1977).

Devijver, P. A., and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, Englewood Cliffs, NJ (1982).

Dietz, W. E., E. L. Kiech, and M. Ali, "Jet and Rocket Engine Fault Diagnosis in Real Time," *J. Neural Network Comp.*, **5** (1989).

Doig, I., *Process Diagnostic Exercises*, CAChE Corp., Austin, TX (1976).

Himmelblau, D. M., *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*, Elsevier, Amsterdam (1978).

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

Huang, W. Y., and R. P. Lippman, "Neural Net and Traditional Classifiers," *IEEE Proc. ICNN*, San Diego (1987).

Isermann, R., "Process Fault Detection Based on Modeling and Estimation Methods—A Survey," *Automatica*, **20**, 387 (1984).

Kohonent, T., *Self Organization and Associative Memories*, Springer-Verlag, Berlin (1984).

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

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

Lieberman, N. P., *Troubleshooting Process Operations*, 2nd ed., Penn Well Books, Tulsa (1985).

Lucas, A. E., and J. Kittler, "A Comparative Study of the Kohonen and Multiedit Neural Network Learning Algorithms," *Proc. First Instl. Conf. on Artificial Neural Networks*, IEE Publ. No. 313, London (1989).

Lyon, R. H., *Machinery Noise and Diagnostics*, Butterworths, Stoneham, MA (1987).

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

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

Oja, E., "Neural Networks, Principal Components, and Subspaces," *Int. J. Neural Systems*, **1**, 61 (1989).

Pan, L. F., *Failure Diagnosis and Performance Monitoring*, Marcel Dekker, New York (1981).

Patton, R., P. Frank, and R. Clark, eds., *Fault Diagnosis in Dynamic Systems*, Prentice Hall, Englewood Cliffs, NJ (1989).

Sacks, R., and S. R. Liberty, *Rational Fault Analysis*, Marcel Dekker, New York (1974).

Ungar, L. H., B. A. Powell, and S. N. Kamens, "Adaptive Networks for Fault Diagnosis and Process Control," San Francisco Meeting, AIChE (Nov., 1989).

Venkatasubramanian, V., and K. Chan, "A Neural Network Methodology for Process Fault Detection," *AIChE J.*, **35**, 1993 (1989).

Venkatasubramanian, V., R. Vaidyanathan, and Y. Yamamoto, "An Analysis of the Learning, Recall, and Generalization Characteristics of Neural Networks for Process Fault Detection," *Comp. Chem. Eng.*, **14**, 699 (1990).

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

Watanabe, K., and D. M. Himmelblau, "Fault Diagnosis in Nonlinear Chemical Processes: II. Application to a Chemical Reactor," *AIChE J.*, **29**, 250 (1983b).

Watanabe, K., and D. M. Himmelblau, "Incipient Fault Diagnosis in Nonlinear Chemical Processes with Multiple Causes of Faults," *Chem. Eng. Sci.*, **39**, 491 (1984).

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

Manuscript received Apr. 3, 1990, and revision received Oct. 23, 1990.