Diagnosis of Multiple Simultaneous Fault via Hierarchical Artificial Neural Networks

Kajiro Watanabe, Seiichi Hirota, and Liya Hou
Dept. of Instrument and Control Engineering, College of Engineering Hosei University, Tokyo, 184 Japan

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

We discuss a new type of macroarchitecture of neural networks called a HANN and how to train it for fault diagnosis given appropriate data patterns. The HANN divides a large number of patterns into many smaller subsets so the classification can be carried out more efficiently via an artificial neural network. One of its advantages is that multiple faults can be detected in new data even if the network is trained with data representing single faults. The use of a HANN is illustrated in fault diagnosis of a chemical reactor.

Introduction

Changes in the physical conditions of process units, control systems or exogenous conditions may lead to what are generically referred to as faults. Faults in the broadest sense include symptoms resulting from physical changes, such as deviations of temperature or pressure from their normal operating range, as well as the physical changes themselves, such as scaling, foaming, leaks, and wear. Even changes in unmeasured process parameters such as heat- or mass-transfer coefficients can be deemed to be faults.

When a process operates under normal conditions, the process parameters are deemed to be at their normal values. If some physical change in the equipment causes deviations from the normal state, the model parameters of the process will deviate from their normal values. Rather than using the measurements (observed variables) to diagnose a faulty condition, here we use the parameters to isolate faults. Because relationships exist among the parameters and measurements of the process, a fault can be diagnosed by making a series of measurements of the process responses and interpreting them in terms of the parameters.

The problem of fault diagnosis can be treated as one of pattern recognition. By making a series of measurements of the process variables, the question is how to detect the faults in the process correctly. In other words, how can the mapping from a measurement space to a fault space (Figure 1) be executed?

A wide variety of techniques have been proposed to detect and diagnose faults including redundant instrumentation, knowledge-based systems, process modeling, statistical tools, digraphs, fault trees, and combinations of these. What is of special interest is incipient fault detection and diagnosis, that is, detection and diagnosis at the very beginning stages of the occurrence of the fault, hopefully prior to the time the fault might be uncovered by an operator. This work concerns the use of artificial neural networks (ANN) to do incipient fault diagnosis for unsteady-state processes in which more than one fault occurs.

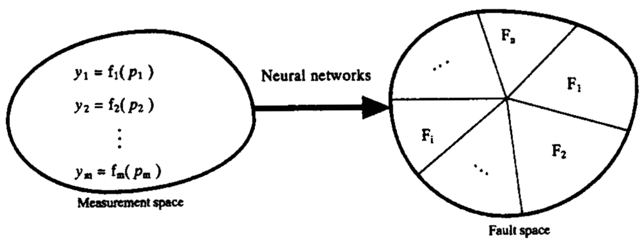

Figure 1. Mapping from a measurement space to a fault space.

One of the advantages of using an ANN for fault diagnosis is that ANN, being nets of basis functions, can map functions quite well so that they can satisfactorily model a complex process whose model is unknown, or whose model is imperfect, at least in the region for which process data are available. The difficulty with diagnosing multiple faults based on the first principles models or state space models is that the process model needs to be almost perfect and extensive calculations are needed. If there are errors in the model, they may be interpreted as faults, thus yielding false alarms and/or preventing faults from being detected. On the other hand, fault diagnosis via a rule-based expert system does require a rule database whose construction is quite time-consuming on the part of process experts, operators, and engineers.

The main disadvantage of ANN for fault diagnosis in practice compared with, say, an expert system is that data for operations in the presence of faults are often quite scarce relative to data for normal operations, so that the use of an ANN alone may not be possible.

To compete with traditional methods of fault detection and diagnosis, ANN must:

1. Involve simple software and hardware implementations
2. Be reliable in the sense of having low rates of type I (false alarms) and type II (lack of detection of a fault) errors.
3. Be able to diagnosis the fault/fault source in real time
4. Be able to ascertain the degree or extent of the fault.

Artificial neural networks, once trained, do inherently fulfill criteria 1, 3, and 4. A critical matter (for any fault diagnosis technique) is the availability of information/data to satisfy criterion 2.

A modest amount of literature exists on the use of ANN to detect and diagnose faults in systems in general. Table 1 lists some recent work in the area of engineering, but considerable literature also exists in psychology, medicine, neurology, and cognitive science.

We have been working on incipient fault diagnosis by using task-specific neural networks. In 1989, Watanabe et al. successfully diagnosed single faults in a reactor. Here we focus on the more general problem of diagnosing multiple faults at an early stage more quickly and efficiently than by other techniques such as the model-based fault diagnosis strategy proposed by Watanabe and Himmelblau (1983). Our intention is to develop task-specific neural networks that are efficient for diagnosing multiple faults and can be easily trained. With this goal in mind, we introduce here a new macroarchitecture of neural networks named hierarchical artificial neural networks (HANN). A HANN is formed in two stages, each of which has a different function. The most distinctive feature of a HANN is that it has the ability to divide a huge set of patterns into a number of smaller sets which in turn can be more easily and efficiently classified. We show how classification of multiple simultaneous faults can be generated from training with fundamental data involving single faults.

In what follows, we discuss the technique of diagnosis, the architecture of the HANN, and the application of HANN to the diagnosis of multiple faults in a chemical reactor.

Table 1. Recent Work on the Use of ANN for Fault Detection and Diagnosis

Authors	Research Topic
Decker (1989)	Application to spacecraft power systems
Dietz et al. (1989)	Application to rocket engine
Hoskins et al. (1991)	Diagnosis of a large chemical plant
Koshijima and Niida (1992)	Meshing of net with expert system
Kramer and Leonard (1990)	Evaluation of back propagation nets
Leonard and Kramer (1990)	Comparison of radical basis function and back propagation nets
Leonard and Kramer (1991)	Use of radial basis function nets
Marko (1991)	Diagnosis of automobile malfunctions
McDuff and Simpson (1990)	F-16 flight line diagnostics
Montgomery (1989)	Use of abductive nets
Naidu et al. (1989)	Detection of sensor failure
Okafor (1991)	Diagnosis of milling machining
Shaw (1990)	Diagnosis of a pump
Sorsa and Koivo (1991)	Evaluation of nets and application to a heat exchanger
Unger and Powell (1988)	Applications to a reactor
Unger et al. (1990)	Steady-state processes
Venkatasubramanian (1989)	Application to catalytic cracking and Chan including multiple faults
Venkatasubramanian et al. (1990)	Reactor distillation column; includes study of multiple faults
Watanabe et al. (1989)	Diagnosis of a chemical reactor
Yamamoto and Venkatasubramanian (1990)	Combined qualitative and quantitative nets
Yao and Zafiriou (1990)	Detection of sensor failure using local receptive fields

Technique of Fault Diagnosis

The proposed procedure for fault diagnosis is discussed with two assumptions: 1. the deviation of the system parameters from their normal values as a result of a fault is slight; 2. all faults occur to roughly the same relative degree of deviation in the process variables. Consider a set of multiple faults corresponding to an AND set in the fault space in Figure 1. For example, for all of the double and triple faults, the AND set is:

$$\{\{F_1, F_2\}, \{F_1, F_3\}, \ldots, \{F_{n-1}, F_n\}, \{F_1, F_2, F_3\}, \ldots, \{F_{n-2}, F_{n-1}, F_n\}\}$$

Given a suitable neural network for diagnosing faults, the degree of success in diagnosis depends on the character of the data used in training the net. If the net is trained to diagnose single faults, how can the net be used to make decisions about multiple faults?

As mentioned previously, the measurements y can be expressed as a function of the parameters p in the process to be diagnosed:

$$y = f(p) \tag{1}$$

From assumption 1, we have:

$$p = p^0 + \Delta p \tag{2}$$

where p^0 is the vector of parameters for normal operating conditions. The measurements y can be described by a set of linear equations in the parameters Δp if Eq. 2 is substituted into Eq. 1:

$$y = y_0 + \Delta y = f(p^0) + A\Delta p \tag{3}$$

where

$$A = \left.\frac{\partial f(p)}{\partial p^T}\right|_{p=p^0} = [a_1 \; a_2 \; \cdots \; a_n]$$

is a Jacobian matrix with constant coefficients. Because $y_0 = f(p^0)$, Δy can be evaluated by the following set of linear equations:

$$\Delta y = a_1 \Delta p_1 + \cdots + a_i \Delta p_i + \cdots + a_n \Delta p_n \tag{4}$$

We can use Eq. 4 as a basis for the classification strategy for multiple faults based on the training data for single faults.

To provide a more concrete illustration of the diagnosis strategy, here are three examples of the expected output deviation in measurements for different fault sets for the AND sets which belong to the fault space shown in Figure 1.

Case 1: single faults

AND set $= \{F_1, F_2, \ldots, F_n\}$

Fault	Deviation
F_1	$\Delta y_1 = a_1 \Delta p_1$
F_2	$\Delta y_2 = a_2 \Delta p_2$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
F_n	$\Delta y_n = a_n \Delta p_n$

Case 2: all of the double faults

AND set $= \{\{F_1, F_2\}, \{F_1, F_3\}, \ldots, \{F_{n-1}, F_n\}\}$

Fault	Deviation
F_1, F_2	$\Delta y_1 + \Delta y_2$
F_1, F_3	$\Delta y_1 + \Delta y_3$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
F_{n-1}, F_n	$\Delta y_{n-1} + \Delta y_n$

Case 3: double and triple faults which include F_1

AND set $= \{\{F_1, F_2\}, \ldots, \{F_1, F_n\}, \{F_1, F_2, F_3\} \ldots, \{F_1, F_{n-1}, F_n\}\}$

Fault	Deviation
F_1, F_2	$\Delta y_1 + \Delta y_2$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
F_1, F_n	$\Delta y_1 + \Delta y_n$
F_1, F_2, F_3	$\Delta y_1 + \Delta y_2 + \Delta y_3$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
F_1, F_{n-1}, F_n	$\Delta y_1 + \Delta y_{n-1} + \Delta y_n$

Hierarchical Artificial Neural Networks

The concept underlying the HANN is to develop an architecture for an ANN that cannot only diagnose faults correctly or at least with a minimum misclassification rate, but also can be trained easily. Such an architecture can be said to be an optimal task-specific macroarchitecture for the problem of diagnosing multiple faults. In training a net to diagnose single faults using deterministic training data, the number of rows and columns of the output decision matrix is equal to the number of faults (plus the normal case), assuming each output node representing a fault is 0 for not active and 1 for active. Examine the matrix in column 4 of Table 2. The number of rows of the pattern input matrix is equal to that of faults (plus the normal case), but the number of columns can differ. Examine the matrix in the third column in Table 2.

Table 2. Training Data for Single Faults

Pattern	Net Inputs	Net Outputs	Fault Diagnosis
0	$\Delta y_1^0 \ldots\ldots \Delta y_m^0$	1 0 0 . . . . . . 0 0 0	Normal
1	$\Delta y_1^1 \ldots\ldots \Delta y_m^1$	0 1 0 . . . . . . 0 0 0	F_1
$\vdots$	$\vdots \quad \vdots$	$\vdots \quad \vdots$	$\vdots$
n	$\Delta y_1^n \ldots\ldots \Delta y_m^n$	0 0 0 . . . . . . 0 0 1	F_n

We will call the optimal task-specific macroarchitecture for single faults the single artificial neural network (SANN). However, in the case of multiple faults, some changes need to be made in both matrices. Take the case of all of the single and double faults in the fault space in Figure 1 given by:

$$\{F_1, F_2, \ldots, F_n, \{F_1, F_2\}, \ldots, \{F_{n-1}, F_n\}\}$$

The number of double faults n_d, and total faults n_t are given by:

$$n_t = 1 + [n + (n-1) + \ldots + 2 + 1] = 1 + [n(n+1)/2]$$

$$n_d = (n-1) + \ldots + 2 + 1 = n(n-1)/2$$

If an SANN net architecture is used to diagnose these sets of faults, $1 + [n(n+1)/2]$ learning patterns (including the normal case) have to be involved. The training matrices can be built up by combining Table 2 with Table 3. The output decision matrix would have two 1s in each row corresponding to the double faults and one 1 corresponding to a single fault. Moreover, the teaching patterns for the double faults are harder to learn than those of single faults. The larger the set of faults (the bigger n), the longer and less certain the training outcome.

To overcome this problem, we propose a hierarchical net, the HANN, which in its simplest form would comprise two connected stages. Examine Figure 2. In the first stage labeled Net^0, all measurements are used as inputs. In the second stage, n identical (or different) nets exist into which are also fed the measurements. Associated with each second stage net is the Boolean output of Net^0. Thus, there are $n+1$ networks. The

Table 3. Training Data for Double Faults

Pattern	Net Inputs	Net Outputs	Fault Diagnosis
$n+1$	$\Delta y_1^{1,2} \ldots\ldots \Delta y_m^{1,2}$	1 1 0 . . . . . . 0 0 0	F_1F_2
$n+2$	$\Delta y_1^{1,3} \ldots\ldots \Delta y_m^{1,3}$	1 0 1 . . . . . . 0 0 0	F_1F_3
$\vdots$	$\vdots \quad \vdots$	$\vdots \quad \vdots$	$\vdots$
n_t	$\Delta y_1^{n-1,n} \ldots\ldots \Delta y_m^{n-1,n}$	0 0 0 . . . . . . 0 1 1	$F_{n-1}F_n$

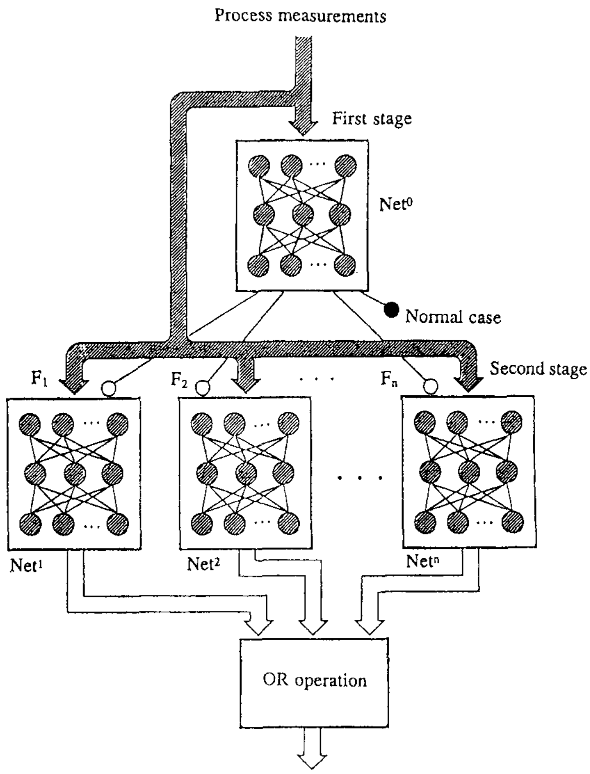

Figure 2. Architecture of a HANN.

faults $\{F_1, F_2, \ldots, F_n, \{F_1, F_2\}, \{F_1, F_3\}, \ldots, \{F_{n-1}, F_n\}\}$ and normal case are divided into $n+1$ sets as in Table 4. $n+1$ learning patterns are involved in the training of Net^0, and n learning patterns are involved in the training of Net^i. The decision matrix of the nets in the second stage used in the training for classification would consist of a single 1 in a row for a single fault, two 1s in a row for double faults, and so on.

The output of the HANN is the result of an OR operation on the outputs of the second stage. The elements of the set used in the OR operation are decided by the output in the first stage of the nets. That is to say, if the ith output node is active in the first stage, the outputs of ith network Net^i in the second stage form the set used in the OR operation.

For example, when faults $\{F_5F_6F_7\}$ occur, and the active outputs from the first stage are $\{F_5F_7\}$, then outputs of Net^5 and Net^7 in the second stage are examined. We find the outputs of the Net^5 are $\{F_5F_6F_7\}$ and the outputs of the Net^7 are $\{F_5F_7\}$. The OR operating on $\{F_5F_6F_7\}$ and $\{F_5F_7\}$ gives the result $\{F_5F_6F_7\}$ for the output of the HANN.

For the cases of single, double, and triple faults, the decisions made by operation of the HANN are as follows:

Rule 1. If the number of nonzero outputs in Net^0 is equal to just 1, then the output of the HANN is deemed to be the output corresponding to the ith output in the first stage.

Rule 2. If the number of nonzero outputs in Net^0 is greater than 1, then the output of the HANN is considered to be the result of an OR operation on several Net^i (the Net^i are those for which the Boolean outputs of $\text{Net}^0 = 1$).

Rule 3. If the number of nonzero outputs in Net^0 is equal to 0, then the result of the HANN is equivalent to deciding: "impossible to be diagnosed."

Both the first stage and the second stage networks are composed of three layers, and can be trained independently by using the backpropagation or another algorithm.

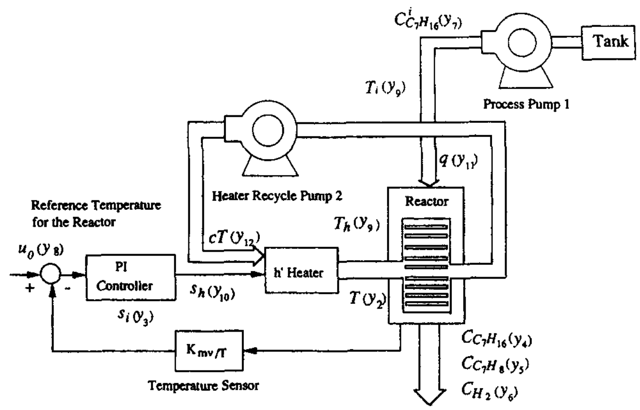

Figure 3. Chemical reactor to be diagnosed.

Application to the Diagnosis of Multiple Faults in a Chemical Reactor

Figure 3 shows a PI feedback-controlled chemical reactor in which heptane is converted catalytically to toluene and which is operated near its steady operating condition (refer to Appendix 1).

This is the same model used by Watanabe and Himmelblau (1983) for fault diagnosis via model based analysis. No catastrophic or abrupt event occurs because any process fault that evolves at an early stage tends to be strongly suppressed and alleviated by the PI controller. 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 is supplied to the heat exchanger via recycle pump 2. The outlet value of the heater is controlled by the PI controller. The controlled variable is the reaction temperature, and the controller regulates the temperature to a specified value. We assumed that the heater control comes from an electronic circuit or a process computer, and we also assumed that the controller is always in a nonfaulty operating mode.

As an example of the application of a HANN, consider seven possible causes of faults:

$F_1 = 0.1\%$ decrease in the command signal u_0
$F_2 = 2.0\%$ decrease in the frequency factor k_0
$F_3 = 0.1\%$ decrease in the activation energy E_a
$F_4 = 2.0\%$ decrease in the heater gain K
$F_5 = 2.0\%$ decrease in the overall heat-transfer coefficient h
$F_6 = 1.0\%$ decrease in the input concentration $C^i_{C_7H_{16}}$
$F_7 = 1.0\%$ decrease in the temperature of the reaction inlet stream T_i

These changes in the parameters provide deviations in the process variables of same degree (2.0%). The fault space is of dimension 7 ($n=7$).

The possible variables in the measurement space are:

y_1 = heater outlet temperature T_h
y_2 = reaction temperature T

Table 4. Networks and Faults Diagnosed in the HANN

Network	Fault Diagnosed				
Net^0	Normal	F_1F_2	...	F_{n-1}	F_n
Net^1	F_1	F_1F_2	...	F_1F_{n-1}	F_1F_n
Net^2	F_2F_1	F_2	...	F_2F_{n-1}	F_2
⋮	⋮	⋮		⋮	⋮
Net^{n-1}	$F_{n-1}F_1$	$F_{n-1}F_2$	...	F_{n-1}	$F_{n-1}F_n$
Net^n	F_nF_1	F_nF_2	...	F_nF_{n-1}	F_n

y_3 = integrator output in the PI controller s_i
y_4 = outlet concentration of $C_{C_7H_{16}}$
y_5 = outlet concentration of $C_{C_7H_8}$
y_6 = outlet concentration of C_{H_2}
y_7 = input concentration of $C^i_{C_7H_{16}}$
y_8 = set point u_0
y_9 = temperature of the reaction inlet stream T_i
y_{10} = driving signal for heater s_h
y_{11} = inlet and outlet volumetric flow rate q
y_{12} = inlet temperature of the heating water cT (c is an efficiency factor that permits the heat losses from the h' heat exchanger and the lines to be included in the problem. We used $c = 0.9$, and multiplied T by c to get y_{12}.)

We want to diagnose the faults with the minimum misclassification possible using a minimum number of measurements. We selected all of the variables involved in the problem to start with, but saw that we needed to minimize the inputs to network by preprocessing of the measurements. In this work, we used cross correlation of the ys to reduce the dimensionality of the measurement space (refer to Appendix 2).

Results of the Diagnosis for the Reactor

Training of the HANN

From the result of the cross-correlation analysis (Table 5), it was found that the variables in

$$\{\Delta C_{H_2}(\Delta y_6),\ \Delta C_{C_7H_8}(\Delta y_5)\},\ \{\Delta u_0(\Delta y_8),\ \Delta T(\Delta y_2)\},$$

$$\{\Delta s_h(\Delta y_{10}),\ \Delta s_i(\Delta y_3)\},\ \{\Delta cT(\Delta y_{12}),\ \Delta T(\Delta y_2)\},$$

$$\{\Delta cT(\Delta y_{12}),\ \Delta u_0(\Delta y_8)\}$$

are dependent. The cross-correlation coefficients are 1.00 between some of the variables because the variables are proportional to each other, such as T and cT, and u_0 and T (Figure 3). Also, the variable $q(\Delta y_{11})$ was not influenced by the faults. Thus we dropped

$$\Delta C_{H_2}(\Delta y_6),\ \Delta u_0(\Delta y_8),\ \Delta s_h(\Delta y_{10}),\ q(\Delta y_{11}),\ \Delta cT(\Delta y_{12}),$$

so that the dimension of the measurement space was reduced from 12 to 7 leaving

$$T_h(\Delta y_1),\ T(\Delta y_2),\ s_i(\Delta y_3),\ C_{C_7H_{16}}(\Delta y_4),$$

$$\Delta C_{C_7H_8}(\Delta y_5),\ C^i_{C_7H_{16}}(\Delta y_7),\ T_i(\Delta y_9),$$

To scale the measurements appropriately, all of the deviations of the process measurements Δy_i were normalized by the standard deviation (selected as 2% of the measurement). Because the coefficient of variation was 2%, even if noise of some degree was added, the classification could be carried out correctly.

The training data in Table 6 were obtained by the reduction and normalization of the process measurements. These data were used to train network Net^0. The inputs were ΔT_h, ΔT, Δs_i, $\Delta C_{C_7H_{16}}$, $\Delta C_{C_7H_8}$, $\Delta C^i_{C_7H_{16}}$, ΔT_i, and the outputs were the fault states and the normal case. For example, the output {0 1 0 0 0 0 0 0} occurred when fault F_1 existed.

Corresponding to the seven faults, seven networks Net^i ($i = 1, 2, \ldots, 7$) existed in the second stage. Each network contained seven inputs and seven outputs. The training data for Net^i were generated by adding to the data in Table 6 the data for two faults as explained earlier. Examine the case of all double faults including F_1 in Table 7. For example, the data of $\{F_1, F_2\}$ in row 2 were generated by adding the data of the row 2 (corresponding to F_1) and row 3 (corresponding in F_2) in Table 6 so the output was {1 1 0 0 0 0 0}.

Both the first- and second-stage nets were feedforward nets containing three layers. For the first stage, the input layer had seven nodes, the hidden layer had thirty nodes, and the output layer had eight nodes. For each net of the second stage, the input layer had seven nodes 7, the hidden layer had thirty nodes, and the output layer had seven nodes.

The objective function for training the HANN was:

$$E = \frac{1}{2}\sum_j (x_{jp} - \hat{x}_{jp})^2$$

where $\hat{x}_{jp}$ was the target input for jth component of the output

Table 5. Cross-correlation ϕ_{ik} among Deviations of the Process Measurements

	T_h	T	s_i	$C_{C_7H_{16}}$	$C_{C_7H_8}$	C_{H_2}	$C^i_{C_7H_{16}}$	u_0	T_i	s_h	q	cT
T_h	1.00	0.48	0.61	−0.35	0.56	0.56	0.22	0.48	−0.11	0.61	—	0.48
T		1.00	0.20	−0.57	0.56	0.56	0.00	1.00	0.00	0.27	—	1.00
s_i			1.00	−0.61	0.31	0.31	0.15	0.20	−0.08	1.00	—	0.20
$C_{C_7H_{16}}$				1.00	−0.49	0.49	0.49	−0.57	0.00	−0.17	—	−0.57
$C_{C_7H_8}$					1.00	1.00	0.52	0.56	0.00	0.31	—	0.56
C_{H_2}						1.00	0.52	0.56	0.00	0.31	—	0.57
$C^i_{C_7H_{16}}$							1.00	0.00	0.00	0.15	—	0.00
u_0								1.00	0.00	0.21	—	1.00
T_i									1.00	−0.08	—	0.21
s_h										1.00	—	0.21
q											—	—
cT												1.00

Table 6. Patterns for Training for Single Faults

Fault	Input Patterns							Output Patterns							
	ΔT_h	ΔT	Δs_i	$\Delta C_{C_7H_{16}}$	$\Delta C_{C_7H_8}$	$\Delta C^i_{C_7H_{16}}$	ΔT_i	1	2	3	4	5	6	7	8
Normal	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1	0	0	0	0	0	0	0
F_1	−1.85	−0.74	−0.23	2.78	−2.78	0.00	0.00	0	1	0	0	0	0	0	0
F_2	−0.83	0.00	−0.17	2.52	−2.52	0.00	0.00	0	0	1	0	0	0	0	0
F_3	0.66	0.00	0.13	−2.03	2.03	0.00	0.00	0	0	0	1	0	0	0	0
F_4	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0	0	0	0	1	0	0	0
F_5	3.04	0.00	0.61	0.00	0.00	0.00	0.00	0	0	0	0	0	1	0	0
F_6	−0.86	0.00	−0.17	−2.38	−2.59	−5.00	0.00	0	0	0	0	0	0	1	0
F_7	0.43	0.00	0.09	0.00	0.00	0.00	−3.00	0	0	0	0	0	0	0	1

Table 7. Patterns for Training for Double Faults Involving F_1

Fault	Input Patterns							Output Patterns						
	ΔT_h	ΔT	Δs_{i_q}	$\Delta C_{C_7H_{16}}$	$\Delta C_{C_7H_8}$	$\Delta C^i_{C_7H_{16}}$	ΔT_i	1	2	3	4	5	6	7
F_1	−1.85	−0.74	−0.23	2.78	−2.78	0.00	0.00	1	0	0	0	0	0	0
F_1F_2	−2.68	−0.74	−0.40	5.30	−5.30	0.00	0.00	1	1	0	0	0	0	0
F_1F_3	−1.19	−0.74	−0.10	0.75	−0.75	0.00	0.00	1	0	1	0	0	0	0
F_1F_4	−1.85	−0.74	0.68	2.78	−2.78	0.00	0.00	1	0	0	1	0	0	0
F_1F_5	1.19	−0.74	0.38	2.78	−2.78	0.00	0.00	1	0	0	0	1	0	0
F_1F_6	−2.71	−0.74	−0.40	0.40	−5.37	−5.00	0.00	1	0	0	0	0	1	0
F_1F_7	−1.42	−0.74	−0.14	−2.78	−2.78	0.00	−3.00	1	0	0	0	0	0	1

pattern for pattern p, x_{jp} was the jth element of the actual output pattern produced by presentation of the input pattern p, and E was the computed least-square error (LSE) for pattern p.

The transfer function for each neuron was:

$$x_j = \frac{1}{1+e^{-\Sigma_i W_{ji} x_i}}$$

where $W_{ji}(t)$ was the weight from the ith to the jth node for the ith iteration.

The algorithm for adjusting the weight was:

$$W_{ji}(t) = W_{ji}(t-1) + \Delta W_{ji}(t)$$

$$\Delta W_{ji}(t) = -\eta \frac{\partial E}{\partial W_{ji}(t)} + \alpha W_{ji}(t-1) \quad (0<\alpha<1)$$

where $\Delta W_{ji}(t)$ was the change to be made to the weight $W_{ji}(t)$, $\partial E/\partial W_{ji}(t)$ was the change in a weight in the error function E, η was the learning rate, and α was a constant which determined the effect of past weight changes on the current direction in weight space.

Table 8. Diagnosing Results of Single Faults

Fault	After 1st Net	After 2nd Stage	After OR Operation	Classification*
F_1	F_1	F_1	F_1	1/1
F_2	F_2	F_2	F_2	1/1
F_3	F_3	F_3	F_3	1/1
F_4	F_4	F_4	F_4	1/1
F_5	F_5	F_5	F_5	1/1
F_6	F_6	F_6	F_6	1/1
F_7	F_7	F_7	F_7	1/1

*Number of faults diagnosed correctly/number of faults occurring.

We set η to 0.1 and α to 0.9. Iteration was continued until the LSE error was smaller than 0.008. The number of iterations required was 6,800 on the average, and the number of training patterns was 29 (1 for the normal case, 7 for the single faults, and 21 for the double faults).

Diagnosis of the faults

Although the training was based on single and double faults, the testing was carried out on single, double, and triple faults. The number of testing patterns was:

Normal case	1
Single faults	7
Double faults	21
Triple faults	35
Total	64

We considered a fault existed in the test when the output of a node in the output layer was greater than 0.5.

Tables 8, 9, and 10 show the results of the diagnoses for single, double, and triple faults, respectively. From the tables, one can observe:

1. All of single faults were diagnosed correctly.
2. For the double faults, when the diagnoses of the first stage were all correct, the HANN yielded the correct diagnoses.
3. For double faults, even though diagnosis results were only partly correct in the first stage, the HANN yielded the correct diagnoses excepting $\{F_1, F_5\}$.
4. For the triple faults, even though diagnosing was 1/3 or 2/3 in the first stage, the HANN diagnosis achieved better than the first stage. Take $\{F_1, F_4, F_7\}$ as an example. The diagnosis of the first stage was just 1/3, but the diagnosis of the HANN was 3/3.
5. Among all the faults, three cases of *impossible to diagnose* ($\{F_2, F_3\}$, $\{F_1, F_2, F_5\}$, $\{F_1, F_5, F_6\}$) occurred.

Table 9. Diagnosing Results of Double Faults

Fault	After 1st Net	After 2nd Stage	After OR Operation	Classification*
F_1F_2	F_1F_2	$F_1F_2 \cup F_1F_2$	F_1F_2	2/2
F_1F_6	F_1F_6	$F_1F_6 \cup F_1F_6$	F_1F_6	2/2
F_2F_4	F_2F_4	$F_2F_4 \cup F_2F_4$	F_2F_4	2/2
F_2F_5	F_2F_5	$F_2F_5 \cup F_2F_5$	F_2F_5	2/2
F_2F_7	F_2F_7	$F_2F_7 \cup F_2F_7$	F_2F_7	2/2
F_3F_4	F_3F_4	$F_3F_4 \cup F_3F_4$	F_3F_4	2/2
F_3F_5	F_3F_5	$F_3F_5 \cup F_3F_5$	F_3F_5	2/2
F_4F_5	F_4F_5	$F_4F_5 \cup F_4F_5$	F_4F_5	2/2
F_4F_6	F_4F_6	$F_4F_6 \cup F_4F_6$	F_4F_6	2/2
F_4F_7	F_4F_7	$F_4F_7 \cup F_4F_7$	F_4F_7	2/2
F_6F_7	F_6F_7	$F_6F_7 \cup F_6F_7$	F_6F_7	2/2
F_1F_3	F_1	F_1F_3	F_1F_3	2/2
F_1F_4	F_1	F_1F_4	F_1F_4	2/2
F_1F_7	F_1	F_1F_7	F_1F_7	2/2
F_2F_6	F_6	F_2F_6	F_2F_6	2/2
F_3F_6	F_6	F_3F_6	F_3F_6	2/2
F_3F_7	F_3	F_3F_7	F_3F_7	2/2
F_5F_6	F_5	F_5F_6	F_5F_6	2/2
F_5F_7	F_5	F_5F_7	F_5F_7	2/2
F_1F_5	$\mathbb{F}_2$	$\mathbb{F}_2F_5$	$\mathbb{F}_2F_5$	1/2
F_2F_3			Impossible	0

*Number of faults diagnosed correctly/number of faults occurring.
$\mathbb{F}_2$ = misclassification.

6. Among all faults, only two instances of misclassification occurred, both for F_2. The likely reason for the misclassification of F_2 is because k_0 is confounded with E_a in the expression for $K(T)$ (see Appendix 1).

Overall the percentages of correct diagnoses were:

Single faults	100%
Double faults	93%
Triple faults	76%

Table 11 lists the results of the classification by the HANN of faults representing different degrees of the fault, and in particular different degrees of deterioration than used in training the HANN. The misclassification is nil. What the table shows is that the HANN was able to generalize (interpolate) well, that is make valid predictions for degrees of faults not included in the training of the HANN.

The question arises as to how well a net trained on small degrees of fault will be able to detect larger faults. In Table 12 we present how well a HANN was able to extrapolate from the region in which training took place. An artificial neural network does not necessarily extrapolate well and is best trained on a wide range of data so that it only has to interpolate. Table 12 lists two kinds of ratios used in diagnosing the results. $R1$

Table 10. Diagnosing Results of Triple Faults

Fault	After 1st Net	After 2nd Stage	After OR Operation	Classification*
$F_2F_4F_5$	$F_2F_4F_5$	$F_2F_4F_5 \cup F_2F_4F_5 \cup F_2F_4F_5$	$F_2F_4F_5$	3/3
$F_3F_4F_5$	$F_3F_4F_5$	$F_3F_4F_5 \cup F_3F_4F_5 \cup F_3F_4F_5$	$F_3F_4F_5$	3/3
$F_1F_6F_7$	$F_1F_6F_7$	$F_1F_6F_7 \cup F_1F_6F_7 \cup F_1F_6F_7$	$F_1F_6F_7$	3/3
$F_3F_5F_7$	$F_3F_5F_7$	$F_3F_5F_7 \cup F_3F_5F_7 \cup F_3F_5F_7$	$F_3F_5F_7$	3/3
$F_2F_4F_7$	F_2F_4	$F_2F_4F_7 \cup F_2F_4F_7$	$F_2F_4F_7$	3/3
$F_2F_5F_7$	F_2F_5	$F_2F_5F_7 \cup F_2F_5F_7$	$F_2F_5F_7$	3/3
$F_4F_5F_7$	F_4F_5	$F_4F_5F_7 \cup F_4F_5F_7$	$F_4F_5F_7$	3/3
$F_4F_6F_7$	F_4F_6	$F_4F_6F_7 \cup F_4F_6F_7$	$F_4F_6F_7$	3/3
$F_2F_6F_7$	F_6F_7	$F_2F_6F_7 \cup F_6F_7$	$F_2F_6F_7$	3/3
$F_3F_4F_6$	F_4F_6	$F_4F_6 \cup F_3F_4F_6$	$F_3F_4F_6$	3/3
$F_4F_5F_6$	F_4F_5	$F_4F_5 \cup F_4F_5F_6$	$F_4F_5F_6$	3/3
$F_5F_6F_7$	F_5F_7	$F_5F_6F_7 \cup F_5F_7$	$F_5F_6F_7$	3/3
$F_1F_4F_7$	F_1	$F_1F_4F_7$	$F_1F_4F_7$	3/3
$F_1F_2F_7$	F_1F_7	$F_1F_7 \cup F_1F_7$	F_1F_7	2/3
$F_2F_4F_6$	F_4F_6	$F_4F_6 \cup F_4F_6$	F_4F_6	2/3
$F_3F_4F_7$	F_3F_4	$F_3F_4 \cup F_3F_4$	F_3F_4	2/3
$F_1F_5F_7$	$\mathbb{F}_2F_7$	$\mathbb{F}_2F_5 \cup \mathbb{F}_2F_5F_7$	$\mathbb{F}_2F_5F_7$	2/3
$F_1F_2F_4$	F_1	F_1F_4	F_1F_4	2/3
$F_1F_2F_6$	F_1	F_1F_2	F_1F_2	2/3
$F_1F_3F_7$	F_1	F_1F_7	F_1F_7	2/3
$F_1F_4F_6$	F_6	F_1F_6	F_1F_6	2/3
$F_2F_5F_6$	F_5	F_5F_6	F_5F_6	2/3
$F_3F_5F_6$	F_5	F_5F_6	F_5F_6	2/3
$F_3F_6F_7$	F_7	F_6F_7	F_6F_7	2/3
$F_1F_2F_3$	F_1	F_1	F_1	1/3
$F_1F_3F_4$	F_4	F_4	F_4	1/3
$F_1F_3F_5$	F_5	F_5	F_5	1/3
$F_1F_3F_6$	F_6	F_6	F_6	1/3
$F_2F_3F_4$	F_4	F_4	F_4	1/3
$F_2F_3F_5$	F_5	F_5	F_5	1/3
$F_2F_3F_6$	F_6	F_6	F_6	1/3
$F_2F_3F_7$	F_7	F_7	F_7	1/3
$F_1F_4F_5$	F_4	$\mathbb{F}_2F_4$	$\mathbb{F}_2F_4$	1/3
$F_1F_2F_5$			impossible	0
$F_1F_5F_6$			impossible	0

*Number of faults diagnosed correctly/number of faults occurring.
$\mathbb{F}_2$ = misclassification.

Table 11. Diagnosing Results of Multiple Faults Combined with Different Deterioration Degrees

Fault	After 1st Net	After 2nd Stage	After OR Operation	Classification*
F_1 (1%) F_2 (0.5%)	F_1	F_1F_2	F_1F_2	2/2
F_1 (1%) F_3 (0.5%)	F_1 F_3	F_1F_3 F_1F_3	F_1F_3	2/2
F_2 (0.5%) F_5 (1%)	F_2 F_5	F_2F_5 $F_2F_5\mathbb{F}_1$	$F_2F_5\mathbb{F}_1$	2/2
F_1 (0.5%) F_2 (1%) F_5 (0.5%)	N, F_1 F_5	$F_1F_2F_5$ $F_1F_2F_5$	$F_1F_2F_5$	2/2
F_1 (0.5%) F_3 (0.5%) F_4 (0.5%)	N, F_1 F_3 F_4	$F_1F_3F_4$ $F_1F_3F_4$ $F_1F_3F_4\mathbb{F}_5$	$F_1F_3F_4\mathbb{F}_5$	2/2

*Number of faults diagnosed correctly/number of faults occurring.
$\mathbb{F}$ = misclassification.
% in ( ) = deterioration degree detected in the measurements.

is the ratio of the number of faults diagnosed correctly to the number of faults occurring, and $R2$ is the ratio of the number of faults diagnosed correctly to the number of faults diagnosed. Table 12 shows how the classification results degrade with increased degree of fault. At the extremes of the fault scale the measurements are well in the nonlinear region of the model, but the HANN still proves a reasonable diagnosis because of the generalization ability of neural networks. Such results cannot be obtained by linear classifiers. The best thing to do if larger faults are of concern and an ANN is to be used for diagnosis is to train the net on data that includes larger faults.

In this example n was 7, and n_d and n_t for a SANN would be 21 and 28, respectively, while n_d and n_t of the HANN just were 6 and 7, respectively. As a result, when we used a SANN, the training for the SANN Table 13 took about 4 times as long as the training for the HANN to achieve the same degree of precision in the classification. The number of single faults n in this reactor example was not very large. Suppose n was a big number, how do the nets compare?

Table 13 compares the HANN with the SANN on a number of learning patterns required for training. From Table 13 we can see that with n increasing, n_d and n_t of the SANN rapidly increase, whereas n_d and n_t of the HANN do not. For example, when n is equal to 20, n_t of the SANN becomes 210, which is about 10 times as large as that of the HANN. This fact lets us conclude that the HANN is more efficient for the case of diagnosing a large number of multiple faults.

We also examined a multiple fault set that contained all the single, double, and triple faults corresponding to 7 single faults. The total number of the faults was 63 (single: 7, double: 21, triple: 35).

Table 12. Diagnosing Percentage of Double Faults with Different Deterioration Degrees

Fault degree	2%	3%	4%	5%	6%	12%	20%
$R1$ (%)	93	93	90	93	90	79	79
$R2$ (%)	95	95	86	79	79	72	70

$R1$ = number of faults diagnosed correctly/number of faults occurring.
$R2$ = number of faults diagnosed correctly/number of faults diagnosed.

Table 13. Comparison of a HANN with a SANN as to the Number of Patterns Required for Training

	SANN		HANN	
n	n_d	n_t	n_d	n_t
2	1	3	1	2
3	3	6	2	3
⋮	⋮	⋮	⋮	⋮
7	21	28	6	7
⋮	⋮	⋮	⋮	⋮
20	190	210	19	20
⋮	⋮	⋮	⋮	⋮

$$\{F_1, \ldots, F_n, F_1F_2, \ldots, F_1F_n, F_1F_2F_3, \ldots, F_{n-2}F_{n-1}F_n\}$$

We attempted to train a single network in which the input layer had seven nodes, the hidden layer had thirty nodes, and the output layer had eight nodes to diagnose these multiple faults. Activation of several nodes would denote multiple faults, but we could not complete the training to a reasonably small error.

Conclusions

We have described a new type of macroarchitecture for neural networks called HANN that can be used for diagnosing multiple simultaneous faults. HANN divides a large number of patterns into many smaller subsets so the training of the networks and the classification of the faults can be carried out more efficiently.

From an example of fault diagnosis using a HANN for diagnosis of a chemical reactor, we have seen that:

1. HANN could correctly classify seven single faults, 21 double faults, and 35 triple faults.
2. Inferences concerning multiple simultaneous faults could be generated from only fundamental data about single faults.
3. A HANN could be trained more easily than a nonhierarchical network for the same purpose.

Notation

F_i = ith fault ($i = 1, 2, \ldots, n$)
F = n-dimensional fault vector
n = number of single faults
n_d = number of double fault combinations
n_t = number of total fault combinations
Net^0 = network of the first stage in a HANN
Net^i = ith network of the second stage in a HANN
p_i = ith parameter
p_i^0 = ith parameter in normal conditions
Δp_i = deviation of p_i
$\mathbf{p}$ = parameter vector
$\mathbf{p}^0$ = parameter vector for normal conditions
$\Delta \mathbf{p}$ = deviation of $\mathbf{p}$
y_j = jth process measurement
$y_j^{i,k}$ = jth process measurement corresponding to F_i and F_k
y_j^0 = jth process measurement in normal conditions
Δy_j = deviation of y_j
$\Delta y_j^{i,k}$ = deviation of $y_j^{i,k}$
$\mathbf{y}$ = m-dimensional measurement vector
$\mathbf{y}_0$ = m-dimensional measurement vector for normal conditions
$\Delta \mathbf{y}$ = deviation of $\mathbf{y}$

$\cup$ = OR operation
$\hat{x}_{jp}$ = target input for jth component of the output pattern for pattern p
x_{jp} = jth element of the actual output pattern produced by the presentation of the input pattern p
E = computed least-square error
t = number of the calculation
$W_{ji}(t)$ = weight from the ith node to the jth node for tth calculation
$\Delta W_{ji}(t)$ = change to be made to the weight $W_{ji}(t)$
α = constant which determines the effect of past weight change on the current direction in weight space
$\partial E/(\partial W_{ji}(t))$ = change in weight on the error function E
η = learning rate

Literature Cited

Decker, D. K., "Some Proposed Applications of Artificial Neural Systems to Spacecraft Power Subsystem," *Proc. IJCNN*, San Diego, 219 (1989).

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

Hoskins, J. C., K. M. Kaliyur, and D. M. Himmelblau, "Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks," *AIChE J.*, **37**, 137 (1991).

Koshijima, I., and K. Niida, "Neural Network Approach to Fault Detection Under Unsteady State Operations," *Proc. IFAC*, Delaware, 174 (1992).

Kramer, M. A., and J. A. Leonard, "Diagnosis Using Back Propagation Neural Networks—Analysis and Criticism," *Comput. Chem. Eng.*, **14**, 1323 (1990).

Leonard, J. A., and M. A. Kramer, "Classifying Process Behavior with Neural Networks: Strategies for Improved Training and Generalization," *Proc. ACC*, 2478 (1990).

Leonard, J. A., and M. A. Kramer, "Radial Basis Function Networks for Classifying Process Faults," *IEEE Control System*, **11**, 31 (1991).

Marks, K., "Neural Network Applications in the Automotive Industry: Diagnosis and Control of Vehicle Systems," *Conf. on Neural Network Application for Manufacturing* (1991).

McDuff, R. J., and P. K. Simpson, "An Adaptive Resonance Diagnostic System," *J. Neural Network Computing*, **2**, 19 (Summer, 1990).

Montgomery, G. J., *Where Do Neural Network Methods Apply in Diagnostics?*, Abtech Corp., Charlottesville, VA (1989).

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, 1336 (1989).

Okafor, A. C., "Multisensor Integration Using Neural Networks for On-Live Process Monitoring," *Conf. on Neural Network Applications for Manufacturing*, Novi, MI (Apr., 1991).

Soroa, T., and H. N. Koivo, "Neural Networks in Process Fault Diagnosis," *IEEE Trans. SMC*, **21**, 815 (1991).

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

Unger, L. H., B. A. Powell, and S. N. Kamens, "Adaptive Neural Nets for Fault Diagnosis and Process Control," *Comput. Chem. Eng.*, **14**, 561 (1990).

Venkatasubramanian, V., "Recall and Generalization Performances of Neural Networks for Process Fault Diagnosis," *Chemical Process Control (CPCIV)*, Y. Arkun and W. H. Ray, eds., 647, CACHE Corp., Austin, TX (1991).

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

Venkatasubramanian, V., R. Vaidyanathan, and Y. Yamamoto, "Process Fault Detection and Diagnosis Using Neural Networks: I. Steady State Processes," *Comput. Chem. Eng.*, **14**, 699 (1990).

Watanabe, K., and D. M. Himmelblau, "Incipient Fault Diagnosis of Nonlinear Processes with Multiple Causes of Faults," *Chem. Eng. Sci.*, **39**, 491 (1983).

Watanabe, K. et al., "Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks," *AIChE J.*, **35**, 1803 (1989).

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

Yamamoto, Y., and V. Venkatasubramanian, "Integrated Approach Using Neural Networks for Fault Detection and Diagnosis," *Proc. IJCNN*, Vol. I, 317 (1990).

Yao, S. C., and E. Zafirious, "Control System Sensor Failure Detection via Networks of Localized Receptive Fields," *Proc. ACC*, 2472 (1990).

Appendix 1: Reactor Model

Variables, constants, and value in the normal state

State Variables
T = reaction temperature, 740.0 K
$C_{C_7H_8}$ = outlet concentration of C_7H_8, 524.0 mol/m^3
C_{H_2} = outlet concentration of H_2, 2,097.0 mol/m^3
$C_{C_7H_{16}}$ = outlet concentration of C_7H_{16}, 476.0 mol/m^3
T_h = heater outlet temperature, 889.0 K
s_i = output of the integrator in PI controller, 223.0 mV
Intermediate Variable
s_h = driving signal for the heater, 223.0 mV
Process Inputs
T_i = temperature of the reaction inlet stream, 300.0 K
$C^i_{C_7H_{16}}$ = inlet concentration of C_7H_{16}, 1,000.0 mol/m^3
cT = inlet temperature of the heating water, 0.9×740.0 K
c = efficiency of h' heater, 0.9
Reactor
k_0 = frequency, 5.01×10^8 1/h
E_a = activation energy, 1.369×10^5 J/mol
R = gas constant, 8.319 J/mol·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/mol
C_p = specific heat, 490.7 J/mol·K
ρ = density, 593.0 mol/m^3
a = area of heat exchange, 10.0 m^3
h = overall heat-transfer coefficient, 6.05×10^5 J/m^2·h·K
q = inlet and outlet volumetric flow rate, 3.0 m^3/h
V = effective reactor volume, 30.0 m^3
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 Controller
K_c = proportional gain, 20.0
T_i^* = integrator coefficient, 0.3 h
$K_{m_v/T}$ = gain of temperature to mV transducer, 1.0 mV/K
u_0 = set point, 740.0 mV

Equations

Reactor Energy Balance

$$K(T) = k_0 e^{(-E_a/RT)}$$

$$\frac{dT}{dt} = \frac{q}{V}(T_i - T) - \frac{\Delta H}{\rho C_p} K \left[T C_{C_7H_{16}} + \frac{ah}{\rho C_p V}(T_h - T) \right], \qquad T(0) = T_0$$

Reactor Mass Balances

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

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

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

Heater Energy Balance

$$\frac{dT_h}{dt}=\frac{1}{\tau}(cT-T_h)+\frac{K}{\tau}s_h,\ T_h(0)=T^0_h$$

$$\frac{ds_i}{dt}=\frac{K_c}{T^*_i}(u_c-K_{mv/T}T),\ s_i(0)=s^0_i$$

$$s_h=K_c(u_c-K_{mv/T}T)+s_i$$

Appendix 2

To minimize the network size, it is always wise to reduce the redundant measurements in so far as possible. Let: Δy^i_j be deviation of the ith process measurement caused by a change in parameter pj which in turn is caused by fault j; Δy_j be n-dimensional deviations vector of the jth measurement due to all (n) presumed faults; and ϕ_{jk} be cross-correlation between Δy_j and Δy_k.

The deviations of some process measurements due to the occurrence of faults may be dependent on the deviations of the other measurements. We reduced the number of process measurements used as input to the HANN by dropping those measurements significantly dependent on others. This reduction does not degrade the accuracy of the diagnosis, but it can reduce the size of the neural network for diagnosis, and hence the training time.

Define an ith measurement vector whose components are the deviations to the presumed faults:

$$\Delta \mathbf{y}^i=[\Delta y^i_1,\ \Delta y^i_2,\ \ldots\ \Delta y^i_j,\ \ldots\ \Delta y^i_m]^T$$

and let the cross correlation of Δy_j and Δy_k be ϕ_{jk}
where $\langle \Delta y^i \Delta y^k \rangle$ is the inner product of Δy^i and Δy^k, $|\Delta y^i|$ and $|\Delta y^k|$ are the norms of Δy^i and Δy^k, respectively, and θ is a threshold determined by the noise level of the measurements.

If $\phi_{jk}>\theta$ and $|\Delta y^i|>|\Delta y^k|$ for all k, then measurement i is retained and the measurement k is dropped.

Manuscript received Mar. 29, 1993, and revision received Aug. 16, 1993.