Diagnostic Model Processor: Using Deep Knowledge for Process Fault Diagnosis

Many recent attempts to use expert systems for process fault diagnosis have included information derived from deep knowledge. This information is generally implemented as a rule-based expert system. Drawbacks of this architecture are a lack of generality, poor handling of novel situations, and a lack of transparency. An algorithm called the diagnostic model processor is introduced; it uses the satisfaction of model equations from process plants to arrive at the most likely fault condition. The method is generalized by the process model and diagnostic methodology being separated. The architecture addresses each of the shortcomings discussed. Experiments show that the methodology is capable of correctly identifying fault situations. Furthermore, information is derived from an a priori analysis technique, which is used to show the degree to which different faults can be discriminated based on the model equations available. The results of this analysis add further insight into the diagnoses provided by the diagnostic model processor.

Thomas F. Petti
Jim Klein
Prasad S. Dhurjati
Department of Chemical Engineering
University of Delaware
Newark, DE 19716

Introduction

The ability of computers to rapidly correlate large amounts of data makes them well suited to the task of chemical process fault detection and diagnosis. However, their use in this area has been extremely modest. Fault detection is still primarily done by plant operators who watch for range violations of either measured or estimated quantities. Diagnosis of the fault is also left to the operators, who rely almost exclusively on their experience with the process under similar situations. Process models are consulted only to diagnose more complex problems. The use of computers for detection and diagnosis is warranted by the increasing complexity of modern chemical plants. Safety issues that require fault situations to be diagnosed quickly and reliably make it difficult to depend solely on human operators.

In recent years, numerous attempts have been made to apply knowledge-based expert systems to the tasks of process fault detection and diagnosis. The various efforts have included many types of knowledge. Since the terminology is used somewhat differently by various investigators, the definitions that will be used for the types of knowledge are as follows:

Shallow knowledge: direct relation between symptoms and causes without the use of system model

Deep knowledge: a process model in a mathematical form

Compiled knowledge: rules formed from information derived from deep knowledge

Rule-based knowledge: rules derived from shallow and compiled knowledge

In general, expert systems for diagnosis have used rule-based knowledge. Process fault diagnosis systems that use rule-based knowledge have been proposed by Chester et al. (1984), Kramer and Palowitch (1985), and Ramesh et al. (1988). The main problems with these types of systems is that they often prove difficult to maintain and validate, they contain a great deal of process-specific knowledge, and they are likely to fail under unanticipated circumstances (Venkatasubramanian and Dhurjati, 1987). Unlike diagnostic expert systems for other applications—for example MYCIN, which is used for medical diagnosis (Shortliffe, 1976)—systems for chemical process domains require the ability to accommodate a new or changing process; in other words, rule bases need to be modified frequently and rapidly. As rules are generally written manually, this is not possible. One solution would be to generate the rules automatically; however, if the rules contain both deep and shallow knowledge, automatic rule formulation would be difficult.

Correspondence concerning this paper should be addressed to P. S. Dhurjati.

The availability of models for chemical processes allows for a more complete representation of the process. Including deep knowledge can help identify fault situations for which there may be no shallow information or contradictory diagnoses. Various attempts to include deep knowledge in a process diagnostic expert system have been made by Dhurjati et al. (1987), Kramer (1987a), and Rich and Venkatasubramanian (1987).

Dhurjati's approach in FALCON uses information about failed model equations in production rules. This system is discussed in more detail in the next section; however, since the deep knowledge is compiled and interspersed with shallow knowledge, it suffers from the drawbacks already discussed. Additionally, the information is put together in an ad hoc fashion to form rules, and the results from the equations are strictly Boolean.

Kramer (1987b) suggests a general diagnostic framework (GDF) that combines causal relationships with mathematical models to represent the deep knowledge of a process. This framework is a generalized representation of all model-based strategies, including the methodology presented here. The subset of the GDF that is based on the satisfaction of model equations is the method of governing equations (Kramer, 1987a). This method is more functional than the method of rule formulation used in FALCON because the model equations are systematically considered to form rules based on their dependence on possible fault situations. The knowledge base construction is more straightforward and mathematically justified; the rule base uses non-Boolean reasoning to preserve more of the information from the model equations. The rule base, although derived from deep knowledge, is still compiled and therefore lacks generality. The diagnosis is also limited to single-cause situations. Kramer's rule formulation methodology is further compared to our approach in a later section.

Venkatasubramanian and Rich (1988; Rich and Venkatasubramanian, 1989) have developed the MODEX environment in which a two-tier object-oriented framework is used to incorporate deep knowledge reasoning and shallow knowledge reasoning. The deep knowledge tier uses process model equations and confluences for each subsystem to diagnoses faults in process equipment (Rich and Venkatasubramanian, 1987). However, this method cannot be used for detection and requires extensive sensor information, which the plant may not be outfitted to provide. Additionally, the method may require operator interaction and therefore may not be appropriate when a fast diagnosis is necessary; however, the technique seems particularly suited to the investigation of process degradation.

Many researchers in the area of expert systems for process fault diagnosis have come to the conclusion that some combination of deep and rule-based knowledge must be used to compromise between processing efficiency and diagnostic performance. Rule-based reasoning alone leads to the shortcomings discussed above with rule-based systems. However, the use of deep knowledge alone is considered to be too inefficient and time consuming for real-time applications (Venkatasubramanian and Rich, 1988). The common approach is therefore to use rules to reach conclusions first, then follow with deep knowledge reasoning to test the hypothesis or to use deep knowledge when the rule base comes up short. The role of compiled knowledge however must be carefully justified. Humans use it because that type of pattern matching is far more natural and easily accomplished than the mathematical manipulations of deep knowledge. The compiled rules consist of information that is obtainable from the model and just put into a form that is more comprehensible to a human but less representative of the process. Number crunching is time consuming, nonintuitive, and hence a method of last resort. This is not the case for computers, and the need for compiled knowledge may no longer exist. The inclusion of rules however can be justified to quickly handle "easy" problems, to handle information that is not representable in a model (shallow knowledge), and to help hypothesize or discriminate between possible faults to augment a deep knowledge based approach. This paper is primarily concerned with the handling of deep knowledge. The possible uses of rule-based knowledge are mentioned to demonstrate the points that have been discussed.

The paper starts with a discussion of the problems encountered with a rule-based system, the FALCON project. It then presents a detailed discussion of an approach to deep knowledge based fault diagnosis that offers numerous advantages over other existing systems. The method, called the diagnostic model processor, uses an algorithm to examine process model equations to arrive at the most likely fault situations. This strategy has the implication of being completely general in that is is independent of the process aside from the equations contained in the model. It also eliminates the rule base, making the representation of the process significantly more comprehensible and thus easier to build and maintain. Additional features discussed include non-Boolean techniques to make maximum use of the available information by avoiding Boolean round-off. Non-Boolean reasoning also avoids diagnostic instability problems and allows partial failures and degradations to be monitored. Additionally, the sensitivities of the model equations to parameter deviations are used to properly weight evidence in evaluating possible faults. The possibility of multiple fault situations is allowed by the method. Finally, the reasoning approach lends itself readily to the issues of transparency and graceful degradation.

FALCON

In 1984, the University of Delaware in cooperation with the DuPont company and the Foxboro company began a project to determine the feasibility of using expert system technology for process fault analysis. The FALCON project (Fault AnaLysis CONsultant) is a rule-based expert system designed to perform fault diagnosis on a commercial-scale processing plant. FALCON uses approximately 750 production rules based on both deep and shallow knowledge. Many of the details of the project can be found in Dhurjati et al. (1987); the system did operate on-line at the production facility for a few months. This discussion concentrates on the target process and the problems that were revealed as the project developed.

Target process

The process is a section of an adipic acid plant as shown in Figure 1 (simplified). It centers around the reaction within the **LTC** (low-temperature converter). There are two liquid streams, **feed1** and **feed2**, and one gas stream, **feed3**, entering the process. There is also a gas stream, **gas**, and a liquid product stream, **prod**, leaving the process. Reaction and heat removal take place in the **LTC**. Gas-liquid separation is carried out in the separator, **Sep**. The three input flows, **feed1**, **feed2**, and **feed3**, have flow controllers associated with them. The level in the separator is automatically controlled by manipulating the flow of the prod-

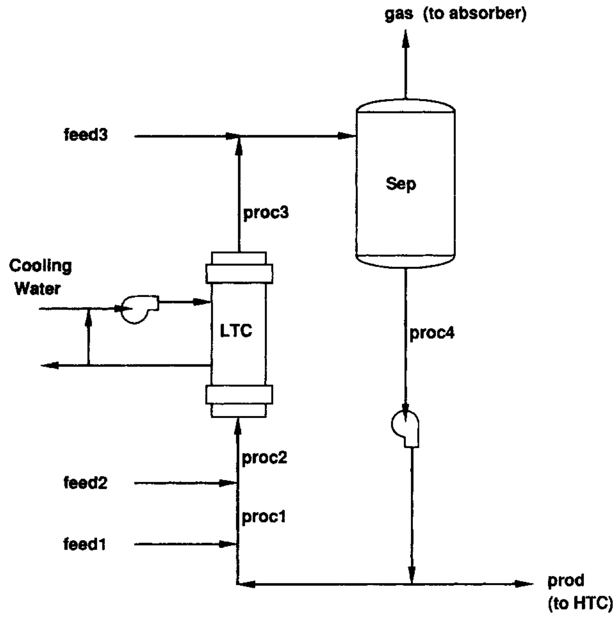


Figure 1. FALCON target process, simplified.

uct stream, **prod.** The **LTC** exit temperature is also controlled using a cascade controller on the cooling water flow.

The information that is made available to FALCON includes all sensor readings, controller set points, and controller outputs. All data are fed directly to FALCON so that operators are not required to enter the information manually and fast diagnoses are possible. Model equations used for deep reasoning include mass, energy, and gas balances; valve curves, which relate controller outputs to flow rates; control laws used in the PI controllers; and heat transfer coefficient and conversion factor correlations. Additionally, trend information is used for the shallow knowledge based reasoning.

A minimum fault set was specified as a precondition of successful operation. These faults include sensor, valve, controller, and pump failures as well as fouling of heat transfer surfaces. Faults such as piping leaks and changes in feed stream concentrations, which are known to occur very infrequently, were not included; other faults, such as loss of feed stream pressure, were not included explicitly but were lumped with faults such as feed stream valves (Dhurjati et al., 1987).

Problems associated with rule-based systems

FALCON is capable of diagnosing all the faults in the minimum fault set and many that were not specified (Dhurjati et al., 1987). However, the degree to which the FALCON project succeeded is not at issue in this treatment. The issue to be addressed here is associated with the shortcomings that are based in the structure of FALCON, namely that of a rule-based expert system. Many of the issues to be covered are discussed in Venkatasubramanian and Dhurjati (1987).

The primary problems with FALCON and other rule-based expert systems of this kind are caused by the complex and subtle interactions of the production rules. The process model is buried in the knowledge base in such a manner that it is impossible to accommodate a new or changing process without repeating an exhaustive verification procedure. This rule structure makes FALCON completely process specific and useless for a new or modified process.

An example of a FALCON rule is as follows:

IF:	mass balance is high
AND:	**feed2** valve curve is high; calculated value agrees with valve curve.
NOT:	**feed1** flow sensor is high; level sensor is stuck; **prod** flow sensor is low; expecting "strange behavior."
THEN:	**feed2** flow sensor is high.

In each rule many parts of the process must be considered along with shallow knowledge variables. In a system that contains hundreds of rules, it becomes obvious that slight changes in the process would require major changes of the knowledge base. The fact that much of the knowledge base would need to be altered gives rise to serious concerns about the validity of the knowledge base should process changes occur. Extensive testing and evaluation of the system's diagnostic ability must be done again to ensure that the analyzer is still competent.

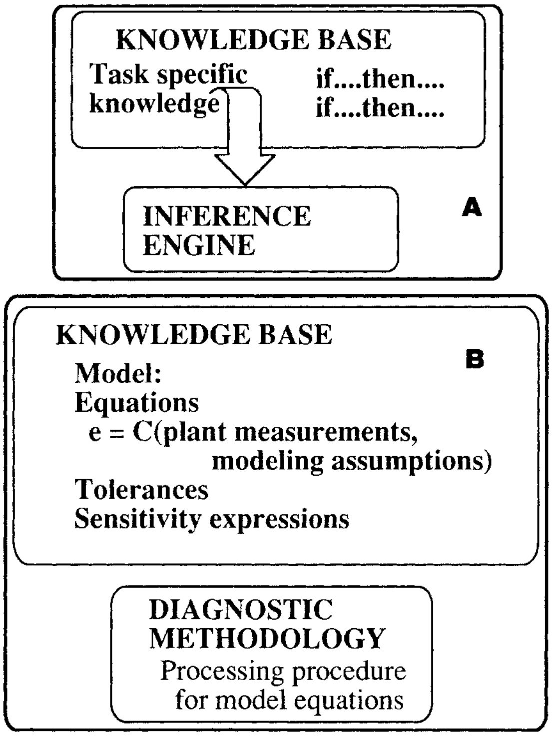
Further problems associated with FALCON involve the handling of novel situations and graceful degradation. Should the FALCON system encounter a process condition that was not foreseen by the system developers, either an incorrect diagnosis or no diagnosis would result. Since it was specified that no analysis be reported until it was verified, the latter would be more likely. However, an incorrect diagnosis could still be given, should the partial evidence of a novel fault fit the pattern of evidence of an anticipated fault.

The problems just described have inspired a more algorithmic approach to the processing of deep knowledge. The following sections describe a methodology that attempts to provide an answer for all the shortcomings found in a rule-based system. The goal is to maximize the information that can be used from a given set of well-formulated model equations.

Diagnostic Model Processor

In an attempt to rectify some of the problems discussed, a more generalized approach to fault analysis has been undertaken. To separate all process-specific model knowledge from the task-specific diagnostic methodology, the knowledge base is built from the process model in a mathematical form and not as rules. This is accomplished by moving the task-specific information into the part of the system that in a traditional expert system would be the inference engine. The process model is now easier to represent at the expense of a general inference engine, which is now only capable of fault diagnosis. This is a superior architecture however, since the process-specific model is isolated and is easily built and modified in this form. Figure 2 shows the architecture of diagnostic expert systems.

The knowledge base consists of a vector of model equations. These statements are merely listed as a series of governing equations that describe the process. Associated with each model equation are tolerance limits, which give an indication of when the equation is no longer representative of the process. Also needed is an expression for determining the sensitivity of each model equation to various parameters. The model equations are written in a form such that they ideally equal zero. In actuality, process noise and model inaccuracies prevent them from equaling zero; the discrepancy is called the residual. The tolerance limits are the expected upper and lower values of the residuals for which the equation is considered to be satisfied. Also associated with each model equation is a set of assumptions



a. Compiled knowledge-based system; process- and task-specific knowledge base with general inference engine

b. Diagnostic model processor; process-specific deep knowledge base with task-specific diagnostic methodlogy

Figure 2. System architecture for diagnostic expert systems.

which if satisfied guarantee the satisfaction of the equation. A simple example of the formulation of a model equation and the associated assumptions (possible faults) is illustrated for the mixing tee in Figure 3. A model equation can be written for the mass balance about the tee:

$$e_1 = (\rho_1 F_1) + (\rho_2 F_2) - (\rho_3 F_3)$$

The assumptions associated with this equation would be:

1. The sensors are functioning and correct
2. The fluids have the expected densities
3. There are no piping leaks

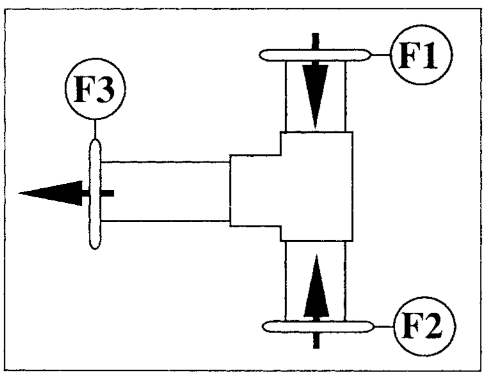


Figure 3. Mixing tee example for model equation formulation.

Some of these assumptions are explicit in the model equation, such as correct sensor readings, and some are implicit, such as the fact that there are no piping leaks. Values for the upper and lower tolerances are required and the various sensitivity expressions are needed to relate each model equation to deviations of each assumption. The details of how the algorithm arrives at and uses these values will be described shortly.

The method uses the fact that violation of a model equation indicates that at least one of its associated assumptions is invalid. By examining the direction and extent to which each equation is violated, and by considering the assumptions on which they depend, the most likely failed assumption can be deduced. An assumption that is common to many violated equations is strongly suspect, whereas satisfaction of equations provides evidence that associated assumptions are valid. Multiple faults are considered possible and the algorithm uses the sensitivity value to weight model equations as evidence. The formulation of the process model is very important to ensure the competence of the fault analyzer. Care must be taken to include all the applicable assumptions with the model equations so that the associated faults can be diagnosed. Note also that the model equations that can be used are dependent on the instrumentation of the target process; balances that require unavailable measurements cannot be used. The model equations for the FALCON target process are listed in Table 1; the associated assumptions that are considered are in Table 2. The following section provides the details of how the diagnostic model processor uses this information to diagnose faults.

Table 1. Model Equations for FALCON Target Process*

Ref. No.	Description
e0	Overall mass balance
e1	Overall energy balance
e2	Overall gas balance
e3	**feed1** valve equation
e4	**feed2** valve equation
e5	**gas** valve equation
e6	**prod** valve equation
e7	**feed3** valve equation
e8	**feed1** flow control law
e9	**Sep** level control law
e10	**feed2** flow control law
e11	**feed3** flow control law
e12	Primary water controller constraint
e13	Secondary water controller constraint
e14	**proc4** temperature error constraint
e15	Water inlet temp error constraint
e16	Dynamic energy balance on **LTC** (exit)
e17	Dynamic energy balance on **LTC** (inlet)
e18	Dynamic energy balance on **LTC** (overall)
e19	Make-up water temperature constraint
e20	Water inlet temperature constraint
e21	Circulation pump flow constraint
e22	Heat transfer coefficient correlation
e23	Conversion factor correlation
e24	**proc3** temperature error constraint
e25	**proc2** temperature error constraint

*Figure 1

Table 2. Assumptions (Possible Faults) Considered

Ref. No.	Assumption
a0	**feed1** flow sensor OK
a1	**feed2** flow sensor OK
a2	**feed3** flow sensor OK
a3	**proc1** flow sensor OK
a4	**prod** flow sensor OK
a5	**Sep** level sensor OK
a6	**gas** flow sensor OK
a7	**Sep** pressure sensor OK
a8	Absorber pressure sensor OK
a9	**proc2** temperature sensor OK
a10	**proc3** temperature sensor OK
a11	**proc4** temperature sensor OK
a12	**prod** temperature sensor OK
a13	Water inlet temp sensor OK
a14	Water exit temp sensor OK
a15	Steady state
a16	**feed1** flow valve OK
a17	**feed2** flow valve OK
a18	**gas** flow valve OK
a19	**prod** flow valve OK
a20	**feed3** flow valve OK
a21	**feed1** flow controller OK
a22	**feed2** flow controller OK
a23	**Sep** level controller OK
a24	**feed3** flow controller OK
a25	Primary water flow controller OK
a26	Secondary water flow controller OK
a27	Complete conversion
a28	Water pump OK
a29	Circulation pump OK
a30	No frosting
a31	No fouling

Diagnostic methodology

The diagnostic process starts with the reading of a vector of process data P. All the residuals from the model equations, c, are then calculated,

$$e_j = c_j(P; a) \tag{1}$$

where a is included to indicate that the equations are also dependent on the vector of assumptions. The model residual, e_j, is a measure of the deviation of the jth model equation from "normal" operation. Normal operation is characterized by the set of equation tolerances, τ, which mark the transition of the equation into the violated regime. The manner in which the tolerances are selected is not critical because a non-Boolean approach is used which allows for a gradual change in the classification of the residual from satisfied to violated. Also, this tolerance will be taken into consideration when the sensitivity of a model equation to an assumption is determined.

A metric is then calculated for all model equations that indicates the degree to which the model equations are satisfied: 0 for perfectly satisfied, 1 for severely violated high, and -1 for severely violated low. These values constitute the satisfaction vector, sf, calculated using the belief function of Kramer (1987a). For the jth model equation,

$$sf_j = \frac{(e_j/\tau_j)^n}{1 + (e_j/\tau_j)^n} \tag{2}$$

sf_j is given a positive value for a positive residual, e_j, and a negative value for a negative residual. The curve is a general sigmoidal function with the steepness determined by the constant n; the tolerances τ do not need to be symmetric about the origin; see Figure 4. If the tolerances are not symmetric about the origin, the upper tolerance is used in the calculations for a positive residual e_j, and the lower tolerance is used if the residual is negative.

Once the sf values are calculated, a sensitivity matrix, S, is determined for the relationship between assumptions and model equations. The sensitivity for those assumptions that are explicit in the model equations (sensor readings) is determined as follows: the sensitivity of the jth model equation to the ith assumption,

$$S_{ij} = \frac{\partial c_j / \partial a_i}{|\tau_j|} \tag{3}$$

In other words, the larger the partial derivative of an equation with respect to an assumption, the more sensitive that equation is to deviations of that assumption. Similarly, equations with large tolerances, τ, are less sensitive as they are more difficult to violate. Many model equations are nonlinear in some assumption dependencies; these partials are estimated by linear approximations. If a model equation uses a time derivative to account for dynamics, this is approximated using a backward difference equation. The partial derivative is then estimated using the partial with respect to the rate of change of the quantity. This is only accurate for the duration of one sampling period after the start of the assumption deviation, but it serves as an approximation. Assumptions that are implicit with respect to an equation

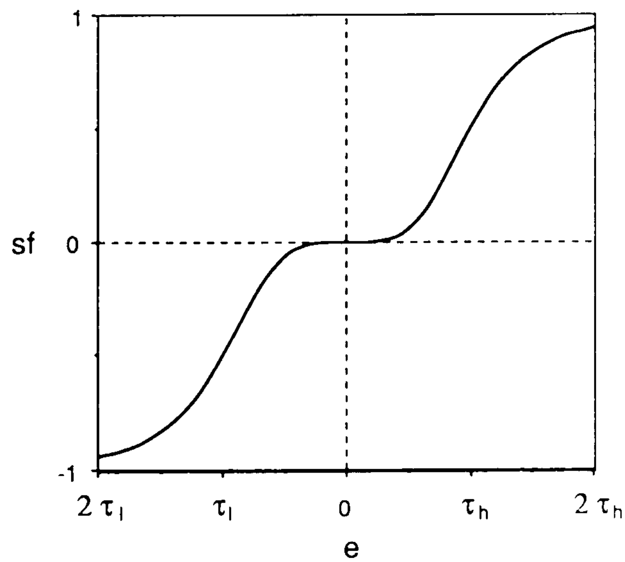


Figure 4. Satisfaction value as a function of equation residual.

High and low tolerance levels are indicated; $n = 4$

(e.g., pump operation) are arbitrarily given a partial derivative equal to 1 unless experience suggests otherwise. Because implicit assumption sensitivities are usually dependent on the magnitude and type of the failure, a better estimate of the partial derivatives is generally not available. This is not crucial because the actual values of the sensitivities are not as important as the relative magnitudes between the model equations. If all equations are given equal partial derivatives for the dependent implicit assumption, the method may not be capable of picking up the failure unless a large failure magnitude is achieved. This may be improved if certain equations, that are known to be more sensitive to an implicit assumption, are given larger partial derivatives. Also, equations have a sensitivity of zero to assumptions that are independent of the equation.

The satisfaction vector, sf, and the sensitivity matrix, S, are then combined to determine the likelihood of a particular assumption violation. A sensitivity weighted average is used to allow for direct contradiction of failures in opposite directions. The failure likelihood F_i of assumption a_i is determined from the equation

$$F_i = \frac{\sum_{j=1}^{N} (S_{ij}\, sf_j)}{\sum_{j=1}^{N} |S_{ij}|} \tag{4}$$

where N is the total number of model equations. The sf_j values of those equations that are most sensitive to deviations of the assumption a_i are weighted the most heavily. Values of F_i that approach 1 indicate a likely condition of assumption a_i failing high; F_i values that go toward -1 indicate a failure low. Equation 4 does not assume that the failure possibilites are mutually exclusive, and therefore does not consider satisfaction values of equations that are independent of an assumption (associated sensitivities are zero). Evidence of the occurrence of one "fault" cannot be used as evidence that another has not occurred. This is important because not all assumptions are associated with infrequently occurring faults, as the term "fault" has been used here to mean any assumption deviation. Some sensor may fail outright while at the same time there exist slight sensor biases and fouling of heat exchanger surfaces. Taking this approach, all possibilities are considered without limiting the search to a single fault explanation of the evidence. This is discussed further in example 2, below, and in the section on fault identifiability.

Secondary model equations

As additional evidence, secondary equations are formed from the model equations previously discussed. Linear combinations are generated which eliminate explicit assumptions from the secondary equation. From a mathematical standpoint, eliminating an assumption between two model equations does not really provide any new information, as a secondary equation is merely a linear combination of two primary equations. However, the use of these equations here is to examine dependent assumptions to diagnose faults. The secondary equation is not dependent on the eliminated assumption, so that should the secondary equation be strongly violated, the evidence would suggest that one of the remaining dependent assumptions is a more likely failure candidate.

Residuals for the secondary equations are formed by combining the jth and kth primary equations, eliminating assumption a_i,

$$e_{jk;i} = e_j - \frac{\partial c_j/\partial a_i}{\partial c_k/\partial a_i} e_k \tag{5}$$

Similarly, secondary tolerances are calculated

$$\tau_{jk;i} = \tau_j + \frac{|\partial c_j/\partial a_i|}{|\partial c_k/\partial a_i|} \tau_k \tag{6}$$

All assumptions that are dependent on the parent equations are dependent on the secondary equations as well, unless they are eliminated. The partials of the secondary equations with respect to the dependent assumptions, a_l,

$$\frac{\partial c_{jk;i}}{\partial a_l} = \frac{\partial c_j}{\partial a_l} - \frac{\partial c_j/\partial a_i}{\partial c_k/\partial a_i}\frac{\partial c_k}{\partial a_l} \tag{7}$$

All the information is now available to calculate the sf and S values for the secondary equations. These can then be included in the calculation of F in Eq. 4. This additional evidence helps in discriminating between deviations in assumptions that appear in the same model equations. Useful information however can only be generated if the parent equations are linearly independent.

Advantages

The major advantage of the architecture of the diagnostic model processor is a true separation between the process model and the diagnostic procedure. This allows for easy changes to the process knowledge to include altered plants or improved equations. Additionally, new processes can be handled by using the design equations associated with them. The non-Boolean reasoning techniques also offer advantages in that they avoid stability problems that may result if equation residuals are bordering near tolerances. The choice of tolerance levels is also less significant and a fault can be monitored as a degradation rather than a Boolean event. The issues of transparency and graceful degradation are addressed in that the model equations, with their dependent assumptions, can always be traced to provide explanations for the conclusions reached. In the event that no diagnosis can be reached, the pattern of violated equations can provide some information that will be useful to the operators. Often, however, a set of possible explanations will result when a single, conclusive diagnosis cannot be made.

An approach that is similar in structure to the diagnostic model processor is the method of governing equations (Kramer, 1987a). This method is used to formulate a rule base that uses non-Boolean certainty factors to calculate a supportability of failure based on the mathematical theory of evidence (Shafer, 1976). To avoid the shortcomings of a rule base, this method of combining certainty factors can be cast in the architecture of the diagnostic model processor. If the absolute value of the satisfaction value, sf, is used as the belief value of failure for each equation, the supportability of assumption i failing can be calculated as follows:

$$q(i) = \left[\prod_{j=1}^{L} |sf_j|\right]\left[\prod_{k=1}^{M} (1 - |sf_k|)\right] \tag{8}$$

where L is the number of equations that depend on assumption i and M is the number of equations that are independent of assumption i. All equations are used in the calculation of each supportability and a no-fault supportability can be calculated by including all the model equation sf's in the second term of Eq 8. This simplified explanation of the method of governing equations considers only the conditions of failed and satisfied assumptions under reliable equation information. A more detailed approach is possible, one that gives specific consideration to the direction (high/low) of each assumption failure based on the belief value of each model equation being violated high or low. Also, an uncertainty factor may be included to indicate the reliability of the information from each equation (Kramer, 1987a). To include this, however, the uncertainty and the direction in which an assumption deviation will tend to violate the model equations must be known beforehand; this information may not always be available.

The shortcomings of this approach are that the method of governing equations considers all model equations in its reasoning, allows for only single-cause explanations, and considers all equations to be equally sensitive to all fault conditions. This last point is equivalent to the assumption that all model equations that are dependent on a failed assumption will become violated and will do so simultaneously. In actuality, less sensitive equations may only fail if the fault becomes very severe, if at all. The diagnostic model processor considers only the equations that are dependent on the assumption when calculating its failure likelihood, F. This makes the diagnosis more robust to poorly formulated equations, reduces the effect of noise in the calculation, and allows for the possibility of noncompeting multiple faults. The sensitivity weighting used in the diagnostic model processor is a more accurate method of classifying the dependent assumptions and interpreting the evidence. By taking the sensitivity into consideration, there is no chance of excludig a fault possibility based on the satisfaction of equations that are insensitive to deviations of the assumption. The sensitivities are also automatically adjusted to changing process conditions (changing partial derivatives) by continually updating the S matrix. Also, the manner in which sensitivities and satisfaction values are combined to arrive at failure likelihoods, Eq 4, allows for direct contradiction of a simultaneous high and low failure. These two faults therefore need not be considered as separate events.

Examples

During the development of the FALCON system, a dynamic simulation of the process was written and verified to generate process data under fault situations. The simulation consists of approximately 270 simultaneous differential equations and over 1,000 parameters. This simulation is used to generate process data for the example fault conditions.

Twenty-six model equtaions are considered in the diagnosis of a possible 31 assumption deviations. The tolerances on these equations were taken to be those used by FALCON that are representative of normal operation. Some of these tolerances are functions of production rate and are updated dynamically. A list of assumptions and model equation is shown in Tables 1 and 2.

It will be clear from the examples that some type of processing of the information furnished by the diagnostic model processor may be necessary before it is presented to the process operators. The diagnosis provided by the algorithm can be further refined by using the information with some rules to interpret the results for presentation. The method so far has been limited to deep knowledge based reasoning, and the addition of shallow information or other refinement techniques may be necessary to improve the performance and discrimination of the diagnosis. An analysis procedure that can be carried out a priori of on-line operation to determine the fault discrimination possible given a set of model equations is described later. This information will be useful in developing the diagnosis refinement methods necessary to build a successful analyzer.

Example 1. The first example is a fault in which the **feed1** flow valve is ramped open. Equation 2 is used to determine the satisfaction values sf with n set equal to 4. Selected sf values over time are plotted in Figure 5a and the F values for a few assumptions are shown in Figure 5b. The values that are not shown do not deviate far from zero; the only equation that

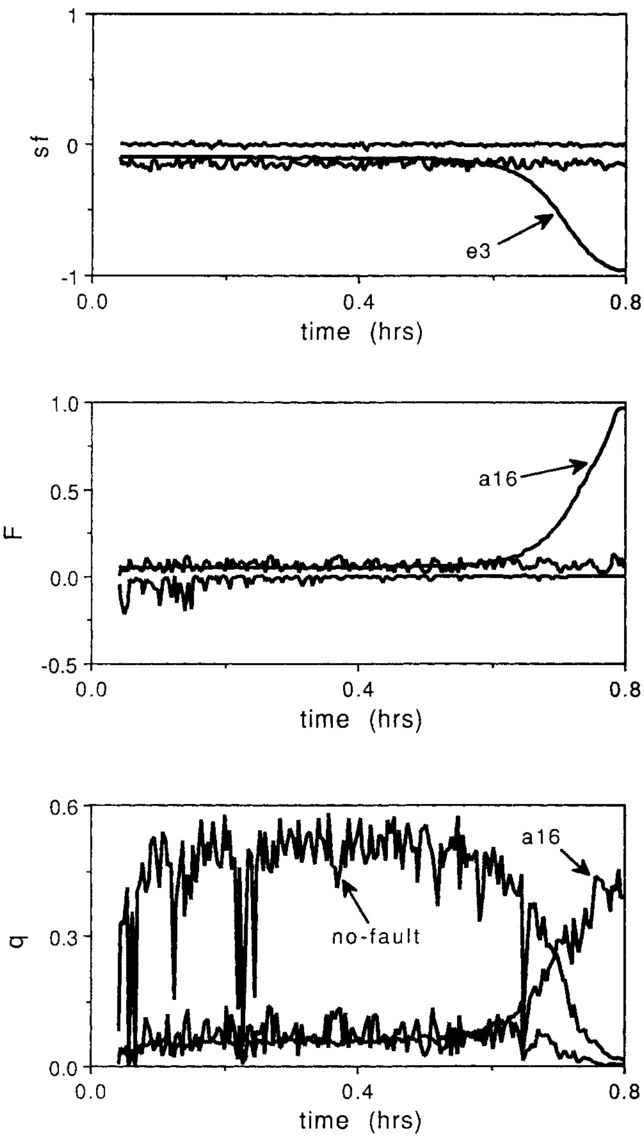


Figure 5. (a) Example 1: satisfaction values. (b) Example 1: failure likelihoods from diagnostic model processor. (c) Example 1: supportability based on method of governing equations.

deviates significantly is the **feed1** valve curve, *e3*. The *F* values show the correct diagnosis of the valve failure high, *a16*. As a comparison, the manner in which certainty factors are combined in the method of governing equations, Eq. 8, is used to arrive at the results in Figure 5c. The method here uses the same tolerances used by FALCON to calculate *sf* rather than the suggested 90% confidence limits (Kramer, 1987a) because they are not available. This figure shows that this method can also successfully pinpoint the fault situation. This however is a simple example in which only one model equation is deviating.

Example 2. A more difficult example is now demonstrated in which the **feed2** flow sensor is faulty and low. Selected *sf* values are shown in Figure 6a The equations that are deviating are a mass balance, *e0*, and the **feed2** valve curve, *e4*. Figure 6b shows the *F* values from the diagnostic model processor. The results indicate high likelihood of the **feed2** sensor failing low, *a1*, but

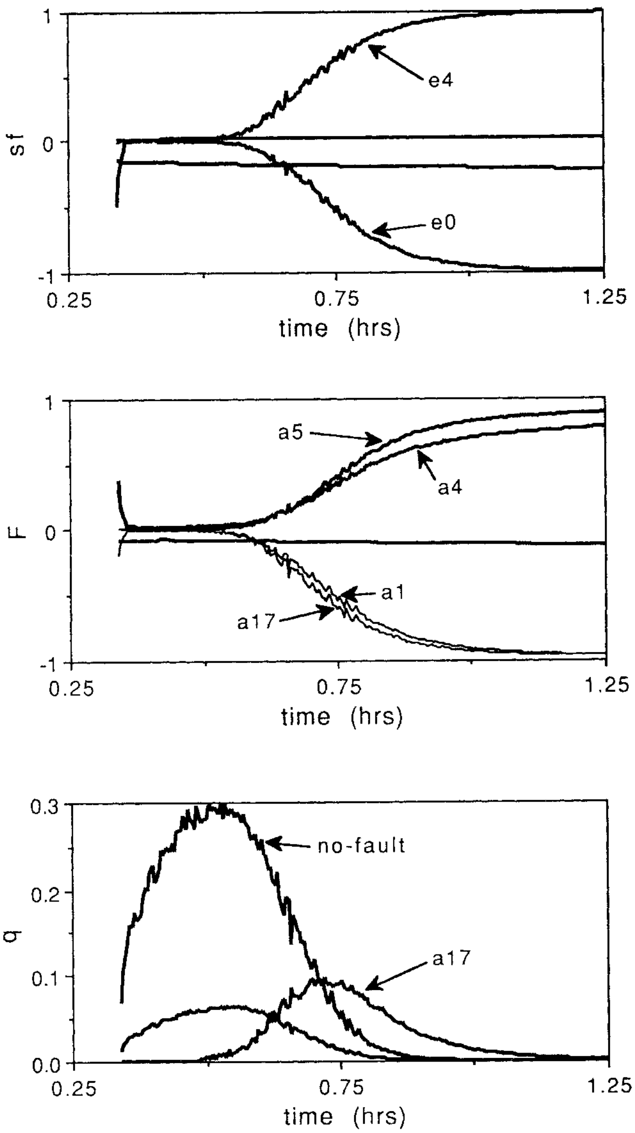


Figure 6. (a) Example 2: satisfaction values. (b) Example 2: failure likelihoods from diagnostic model processor. (c) Example 2: supportability based on method of governing equations.

also high likelihood of the **feed2** valve failing low, *a17*. In addition, significant likelihood is given to the possibility of two other sensors failing high, *a4* and *a5*. Although the correct failure is indicated, the results are not conclusive because of these other failure possibilites. This occurs because the diagnostic model processor does not reduce an assumption's likelihood of failure based on the expected satisfaction of model equations independent of that assumption. This allows for the possibility of noncompeting multiple faults. If the **feed2** valve in this example were truly failing low in addition to the **feed2** sensor, the same pattern of equation violations would result. The diagnostic model processor does not search for a single fault to explain all the evidence, thus allowing for the possibility of multiple faults. It is obvious, however, that some rules to refine the diagnosis are necessary to improve the usefulness of the analyzer. Methods that can be used to improve the performance are considered in the next section.

For comparison, the method of governing equations, Eq. 8, is used to identify the fault in this second example. Figure 6c shows the results obtained. Once again the FALCON tolerances were used. Although it is clear that a fault is occurring (the no-fault indicator drops to zero), the method is unable to pick out the correct fault situation. The reason that the method fails is based on the sensitivity issue discussed earlier. The faulty sensor is a dependent assumption in many model equations and this method expects all of these equations to fail. The sensitivity of these equations to this fault is so low, however, that they do not. The use of this sensitivity in the weighted average method in the diagnostic model processor remedies this problem.

It is clear from the second example that there are cases in which the diagnostic model processor has trouble discriminating between certain fault situations. An analysis tool to determine which faults will cause discrimination problems is discussed below. Also, the use of a single fault hypothesis may be useful to help refine the diagnostic results. As mentioned earlier, the diagnostic model processor relies only on the model equations that depend on an assumption to calculate its failure likelihood. The single fault hypothesis is based on the fact that in the event of a true single fault situation, no model equations should be violated which do not depend on or are insensitive to the suspected assumption, a_i. Additional information may be obtained by including the expected satisfaction of equations independent of the assumption and equations very insensitive to deviations of the assumption in the failure likelihood calculation, Eq. 4. The usefulness of such a hypothesis is a topic of further investigation and caution must be exercised when considering only single assumption deviations. Although multiple fault conditions in a chemical plant may rarely occur, the assumptions associated with a model equation may not be true faults, and therefore multiple deviations may be more likely to occur. An example of an assumption that is not generally associated with a fault is the assumption of steady state required by all balances that do not account for dynamics.

Fault Identifiability

The ability of the diagnostic model processor to identify and discriminate between fault situations is completely dependent on the model equations that are available. An analysis technique has been developed which allows the model equations to be tested a priori of on-line operation to determine the performance of the diagnostic model processor. The technique is based on the significances, *sig*, of the model equations to the diagnosis of the various faults, as defined by the sensitivities:

$$sig_{ij} = \frac{S_{ij}}{\sum_{j=1}^{N} |S_{ij}|} \tag{9}$$

Using these values, a fault situation is assumed to determine which other faults may be confused with the actual pattern of failed model equations.

The comparison matrix

The technique requires a typical vector of process data to calculate the needed sensitivities, Eq. 3, and significances, Eq. 9. The procedure then cycles through each assumption a_i, assuming that the jth equation which is dependent on assumption a_i is violated in proportion to the significance of the equation, sig_{ij}. The rest of the possible assumptions, a_k, are then considered one by one and a comparison value is calculated based on the pattern of equation violations. The comparison values are calculated from the equation:

$$C_{ik} = \sum_{j=1}^{N} \frac{(sig_{kj}\, sig_{ij})}{\max \{sig_i\}} \tag{10}$$

The $max\ \{sig_i\}$ term is used to scale all the assumed equation violations so that the most significant model equation is set to the most severe violation (1 or -1).

The comparison matrix is used to simulate a fault situation and determine how other faults may cause discrimination problems. Large results (over about 0.5) for C_{ik} may indicate that the kth fault will appear as an additional possibility in the diagnosis of the ith fault from the diagnostic model processor. The analysis is approximate because changing process conditions will result in different values for the equation partial derivatives and thus different values for the significances. The analysis is designed to point out potential problems before the analyzer is used so that ways to handle the discrimination problems can be decided early. The use of this matrix is further discussed below.

Examples

The examples used previously can now be revisited to demonstrate the abilities of this analysis technique. A portion of the comparison matrix for this process is shown in Table 3. The typical data vector that was used is from the normal operation of the process. The matrix is organized so that the actual fault situation is listed down in rows, with the faults to be compared across in columns. In the first example, the true fault was successfully identified as the sole failure; see figure 5b. The comparison matrix predicts this behavior. The C values for all comparisons across row *a16* are very small (similar in magnitude to the values shown in Table 3), indicating that a conclusive diagnosis is expected.

Table 3. Portion of Comparison Matrix Calculated from Eq. 10

	a4	a5	a17
a1	−0.56	−0.97	0.99
a16	0.015	0	0

The second example showed that discrimination problems existed when the actual fault situation was the **feed2** flow sensor, *a1* Figure 6b. The comparison matrix indicates values of C over 0.5 in row *a1* for all of the fault situations that resulted in significant likelihood of failure, shown in Table 3. The faults included the **feed2** stream valve, *a17*, as well as the **prod1** stream flow sensor, *a4*, and the separator **Sep** level sensor, *a5*. The analysis successfully predicted the discrimination problems. Note that the signs of the C values correspond to the direction of the inconclusive diagnoses, positive for the same direction (low in Figure 6b) and negative for the opposite direction (high) of the true failure.

Use of the comparison matrix

As demonstrated in the preceding section, the comparison matrix is a useful method of predicting the performance of the diagnostic model processor. The two main uses for the information provided by the analysis include formulating ways of refining the diagnosis of the analyzer and help in designing the needed instrumentation of processes for effective on-line fault diagnosis.

This treatment has considered the use of deep knowledge only. The information supplied by the comparison matrix points out the places where potential diagnoses could be aided by further analysis or some shallow knowledge based rules. The use of these techniques may effectively solve any discrimination problems that exist in the system. An example of a possible refinement rule might be that if the diagnostic model processor indicates high failure likelihood of both a flow sensor and a control valve on the same flow stream, then the more probable fault is the flow sensor. Other more process-specific knowledge could also be included here. The advantages over strictly rule-based systems should still be substantial because the number of these rules should be comparably small. Other refinement techniques are also being investigated, such as the use of a single fault hypothesis, consideration of the relationships between assumptions, and the use of prior probabilities.

The other potential use of this analysis technique is in the design of better instrumentation for effective fault diagnosis. The comparison matrix can be constructed based on model equations from different plant configurations to determine the optimum for diagnosing the largest number of plant failures. Different configurations can also be considered based on the importance of accurately and quickly identifying the most critical fault conditions. An example of improving instrumentation for fault diagnosis follows.

Redundant measurement example

As discussed previously, the comparison matrix indicates that fault discrimination problems exist in the diagnosis of the **feed2** flow sensor. If this is a critical fault condition, instrumentation may be added to help properly identify this fault. The simplest solution is the addition of another sensor to measure this flow. Additional model equations that use the value of this redundant measurement can be included.

Another comparison matrix is constructed which includes the effect of the redundant equations. A selection from this matrix is shown in Table 4. It is clear that a significant reduction in

Table 4. Portion of Comparison Matrix, Including Effects of a Redundant Measurement

	a4	a5	a17
a1	−0.29	−0.49	0.5

comparison values is achieved, indicating that the fault is identifiable; compare with Table 3. To illustrate the improved discrimination, example 2 is examined using the redundant sensor equations. The *sf* values for selected equations are shown in Figure 7a. The equations that show significant deviations are the same as those in Figure 6a, indicating that the redundant model equations are satisfied as expected. The failure likelihoods are plotted in Figure 7b. The failure of the **feed2** flow sensor is now easily identified, as predicted by the comparison matrix, Table 4.

Conclusions

The diagnostic model processor is shown to be an effective method of using deep knowledge in an architecture that is superior to compiled knowledge-based expert systems. The improvements discussed have included issues regarding the generality of the system, a characteristic which is required based on the changes that are continually being made to modern processing plants. The generality of this approach is achieved through the clear separation of process model and diagnostic methodology. The process model is kept as a set of mathematical expressions that describe the process and can easily be improved or modified to accommodate any changes in the process.

Another advantage includes the use of non-Boolean methods to improve the stability of the diagnosis. Additionally, the consideration of the sensitivity of model equations to assumption deviations is effectively used to weight the evidence to arrive at the most likely fault situations. The methodology is transparent in that the diagnosed condition can be traced through the violations of associated model equations. Information that is useful in identifying the fault can always be generated by the method even in the event that it cannot pinpoint a single cause of failure. Furthermore, the algorithm allows for multiple faults by not eliminating hypotheses based on model equations that are independent of the suspected fault.

The examples show that the approach is capable of correctly diagnosing faults. They also show that the diagnosis may report the high likelihood of fault situations that are not occurring. The fault situations must therefore be considered in the light of possible identifiability. The results are useful in that under the analysis technique discussed, methods must be included to handle the expected discrimination problems between critical conditions. The deep-knowledge diagnostic model processor is useful in that all fault situations that would generate the observed evidence are presented. The problem is now one of narrowing the diagnosis through the use of refinement techniques to identify the true state of the process.

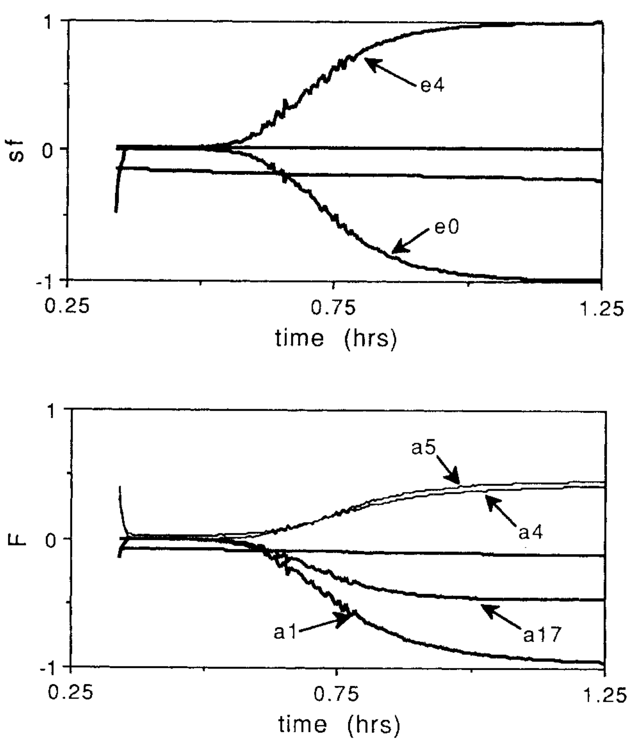


Figure 7. (a) Example 2: satisfaction values of model equations, including redundant measurement. (b) Example 2: failure likelihoods from diagnostic model processor, with redundant measurement.

Acknowledgment

This material is based upon work supported under a National Science Foundation Graduate Fellowship. The authors gratefully acknowledge the support of the National Science Foundation Presidential Young Investigator Program and the Union Carbide Corporation.

Notation

a_i = ith assumption
c_j = jth model equation
C_{ik} = comparison value of kth assumption when the actual failure is the ith assumption, Eq. 10
e_j = residual of jth model equation, Eq. 1
F_i = failure likelihood of ith assumption, Eq. 4
L = number of model equations that are dependent on an assumption
M = number of model equations that are independent of an assumption
n = exponent controlling sigmoidal curve, Eq. 2
N = total number of model equations
P = vector of plant sensor data
$q(i)$ = supportability of assumption i failing based on the method of governing equations, Eq. 8
S_{ij} = sensitivity of jth model equation to the ith assumption, Eq. 3
sf_j = satisfaction value of jth model equation from the sigmoidal curve, eq. 2
sig_{ij} = significance of jth model equation to the diagnosis of the ith assumption failure, Eq. 9

Greek letters

τ_j = tolerance of jth model equation
τ_h = upper tolerance, if not symmetric about the origin
τ_l = lower tolerance, if not symmetric about the origin

Subscripts

i = regarding ith assumption or model equation
ij = regarding relationship between ith assumption (model equation) and jth model equation (assumption)
$jk;i$ = regarding secondary model equation derived by combining the jth and kth model equation, eliminating the ith assumption

Literature Cited

Chester, D. L., D. E. Lamb, and P. S. Dhurjati, "An Expert System Approach to On-line Alarm Analysis in Power and Process Plants," *Comput. Eng.*, **1**, 345 (1984).

Dhurjati, P. S., D. E. Lamb, and D. L. Chester, "Experience in the Development of an Expert System for Fault Diagnosis in a Commercial-scale Chemical Process," *Foundations of Computer Aided Process Operations*, Reklaitis and Spriggs, eds., CACHE/Elsevier, 589 (1987).

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

———, "Expert Systems for Process Fault Diagnosis: A General Framework," *Foundations of Computer Aided Process Operations*, Reklaitis and Spriggs, eds., CACHE/Elsevier, 557 (1987b).

Kramer, M. A., and B. L. Palowitch, Jr., "Expert System and Knowledge-based Approaches to Process Malfunction Diagnosis," *AIChE Nat. Meet.*, Chicago (1985).

Ramesh, T. S., S. K. Shum, and J. F. Davis, "A Structured Framework for Efficient Problem Solving in Diagnostic Expert Systems," *Comput. Chem. Eng.*, **12**, 891 (1988).

Rich, S. H., and V. Venkatasubramanian, "Model-based Reasoning in Diagnostic Expert Systems for Chemical Process Plants," *Comput. Chem. Eng.*, **11**, 111 (1987).

———, "Causality-based Failure-driven Learning in Diagnostic Expert Systems," *AIChE J.*, **35**, 943 (1989).

Shafer, G., *A Mathematical Theory of Evidence*, Princeton Univ. Press (1976).

Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN*, American Elsevier/North Holland, New York (1976).

Venkatasubramanian, V., and P. S. Dhurjati, "An Object-oriented Knowledge Base Representation for the Expert System FALCON," *Foundations of Computer Aided Process Operations*, Reklaitis and Spriggs, eds., CACHE/Elsevier, 701 (1987).

Venkatasubramanian, V., and S. H. Rich, "An Object-oriented Two-tier Architecture for Integrating Compiled and Deep-level Knowledge for Process Diagnosis," *Comput. Chem. Eng.*, **12**, 903 (1988).

Manuscript received Oct. 5, 1989, and revision received Jan. 29, 1990.