A Rule-Based Approach to Fault Diagnosis Using the Signed Directed Graph

Fault diagnosis is the problem of determining the root causes of process upsets. This paper presents a very efficient method of identifying the possible causes of process disturbances using the signed directed graph (digraph) representation of process interactions. The analysis is based on forming logical statements (rules) derived from the process digraph; these are evaluated using on-line data to yield the diagnosis. Evaluation of rule antecedents is more efficient than the previous algorithmic approach of Shiozaki et al. In the rule-based approach, the diagnostic criteria are represented explicitly, not hidden by a complex algorithmic procedure. This allows the diagnostic rules to be tailored to reflect the best available knowledge of plant behavior. The rules generated by this technique can be integrated with other rules on plant operations using an expert systems framework.

M. A. Kramer, B. L. Palowitch, Jr.
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Introduction

In most plants, diagnosis of process upsets is left to the abilities of process operators. When a process alarm is activated, the operator determines the seriousness of the situation and initiates appropriate action. The diagnostic decision is based on the type of alarm, the values of related process variables, and the operator's background, training, and mental model of the plant.

This method of diagnosis has certain disadvantages. The availability of process experts may depend on work shift, employee turnover, vacations, and the like. Operators may be well trained in standard procedures but ill-equipped to handle unusual events. Stress associated with alarm situations can compound the difficulty of decision making. The operator's mental model of the process may be incorrect. These factors make computer-based process monitoring and diagnosis desirable.

The problem of fault diagnosis has been addressed by many authors and is the subject of books by Himmelblau (1978) and Pau (1981). Different process representations distinguish various approaches to this problem. Quantitative approaches involving filtering and estimation have been reviewed by Isermann (1984). Qualitative approaches involving fault trees and related diagrams have been reviewed by Lees (1983). Another qualitative approach, involving the signed directed graph (SDG), has been developed by O'Shima and coworkers (Iri et al., 1979; Tsuge et al., 1985; Shiozaki et al., 1985). Digraph-based methods are attractive because relatively little information is needed to set up the digraph and perform the diagnosis. The approach of O'Shima and coworkers is algorithmic in nature, and involves tracing the possible sources of disturbances using the information stored in the SDG.

The SDG represents pathways of causality in the fault-free process. The nodes of the SDG correspond to state variables, alarm conditions, or failure origins, and the edges (branches) represent the causal influences between the nodes. The direction of deviation of the nodes is represented by signs on the branches, + (−), indicating that the cause and effect variables tend to change in the same (opposite) direction. Variations on the SDG involving multiple time stages and delay times have been proposed by Umeda et al. (1980) and Tsuge et al. (1985). Kokawa et al. (1983) present a diagnostic algorithm incorporating delays, gains, and fault propagation probability in the digraph, but the method is limited to processes without feedback. In the current paper, probability, gain, and time delay information is not used.

In digraph-based methods it is assumed that a single fault, affecting a single node in the SDG (the root node), is the source of all disturbances. It is also assumed that the fault does not alter other causal pathways in the digraph. The fundamental premise of digraph techniques is that cause and effect linkages must connect the fault origin to the observed symptoms of the fault. The diagnosis involves locating all possible disturbance sources (root nodes), given on-line sensor data. The sensor data are classified into three states, high, normal, and low. For the

Correspondence concerning this paper should be addressed to M. A. Kramer.

purpose of this paper, these three states will be represented by $+1$, 0, and -1, respectively.

A methodology for deriving the SDG from process equations is given in Iri et al. (1979). The SDG derived in this fashion has certain limitations not explicit in the previous works, most significantly that the correct diagnosis can be guaranteed only if each variable undergoes no more than one transition between qualitative states during fault propagation. This is because the SDG represents only the initial system response to disturbances. When compensatory or inverse responses cause a variable to change qualitative state more than once during propagation of the fault, a continuous causal pathway from the source node to each disturbed nodes may not exist. An example is given in the Appendix. Although severe, the assumption of single state transitions will be continued in the current work for the purpose of deriving the rule-based format. Further efforts are underway to remove these restrictions.

The algorithm of Shiozaki et al. (1985) for locating the root nodes in the single-state SDG is complex and time-consuming. The example cited in that work, containing 99 nodes and 207 branches is solved in times up to 5 min on a FACOM M-200 computer. This is marginal as far as practical use of the algorithm in real time. The algorithmic technique also has a tendency toward poor diagnostic resolution (multiple unresolvable hypotheses) and sensitivity to alarm thresholds.

The objective of the present work is to show how the digraph can be converted into a concise set of logical rules that are efficient to evaluate and provide a framework for addressing the issue of improved diagnostic resolution. A further benefit of the rule-based format is that the diagnostic rules can be combined with other rules pertaining to plant operations in an expert system. In principle, expert systems approaches are amenable to fuzzy logic, which may help address the problem of alarm threshold sensitivity, although this aspect is not treated here.

Fault Simulation Using the SDG

The diagnosis problem is the inverse or dual of a much simpler problem, namely that of fault modeling. Diagnosis uses a set of observed symptoms to generate a hypothesis regarding the operating state of the plant. Fault modeling predicts the response of the plant, given the operating state. Because of this duality, all methods of diagnosis contain explicity or embedded within them predictions of the symptoms of process faults. The benefit of analyzing the dual problem arises from the relative simplicity of the modeling problem, as compared to the diagnosis problem. Additionally, the procedural aspects of the diagnosis are decoupled from the fault modeling problem and can be ignored, at least initially, in this analysis.

Fault simulation using the digraph involves the formation of a set of directed trees (called simulation trees or interpretations) branching from a given root node. The simulation tree represents a prediction of the dominant pathways of fault propagation, and yields information on the order of events and the direction of deviation of each node connected to the fault origin. Each simulation tree represents one set of routes from the root node to each causally connected node. For a given digraph and a given fault origin, there may be many interpretations of the propagation of the fault. Only one or a small set of these interpretations reflects the real behavior of the plant. However, from the digraph alone, these cannot be distinguished from spurious interpretations.

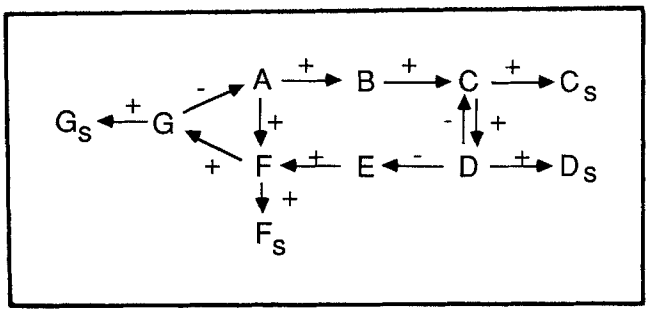


Figure 1. Digraph for example 1.

Formation of simulation trees is most easily demonstrated through an example. The digraph in Figure 1 (example 1) shows an abstract network of interactions between variables. Sensors are represented by separate nodes, indicated by subscript *s*. If measurement is rapid, any working sensor can be lumped with its corresponding variable node, without generating spurious interpretations. If A is selected as the fault origin, then all sensors can be lumped with their corresponding variable nodes. Figure 2 shows the simplified digraph incorporating these assumptions.

In some previous works it has been assumed that a variable and its measurement can be lumped as a single node when the sensor is not an input to a control loop (e.g., Figure 7 in Shiozaki et al., 1985). This lumping can give rise to spurious diagnosis of sensor malfunction, because sensor failures (except in control loops) are not distinguished from actual process malfunctions. The present approach recognizes that lumping is invalid when simulation trees for sensor failure are developed.

The sign of the root node is determined by the fault, and thus any pathways in the digraph leading into the root node are irrelevant. Selecting $A = +1$ as the fault origin means that the fault enters the network at A by perturbing variable A in the positive direction, with large enough magnitude to override the feedback effect from G.

The digraph can be further simplified by removal of unmeasured nodes. These are not of interest in the diagnosis except as potential root nodes. Branches through unmeasured nodes can be replaced by a single branch connecting measured nodes, with the sign on the new branch equal to the product of the signs of the branches it replaces (exceptions to this procedure are explained later). For example 1, the simplified digraph is shown in Figure 3. The root node, even if it is unmeasured, is not removed at this stage.

Since an interpretation is a directed tree, feedback loops must be removed. For uncontrolled variables in negative feedback

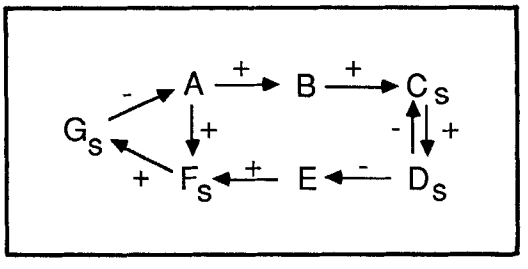


Figure 2. Digraph for example 1 with lumping of sensor nodes.

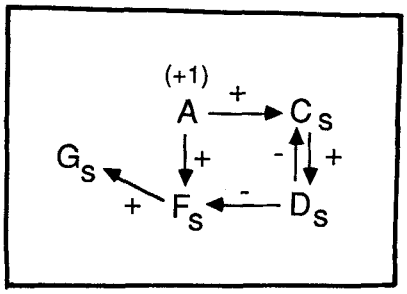


Figure 3. Reduced digraph for example 1, $A = +1$ as root node.

loops, the assumption of single state transitions implies that negative feedback generated by propagation of a disturbance through an uncontrolled node can neither completely compensate nor override the initial disturbance. This implies that the feedback loop from D to C can be ignored, because under the assumption of single transitions, C cannot be returned to normal following the initial disturbance. Control loops must be handled differently, because controlled variables can pass disturbances without significant deviation. It is assumed in this example that feedback loops are not control loops.

The results, shown in Figure 4, are two simulation trees rooted at A. These correspond to two ways in which the disturbance can be transmitted to F: directly from A, or through C and D. We will call these interpretations I and II, respectively. In reality, F is influenced by A and D simultaneously. However, because quantitative information is lacking, the relative magnitudes of these effects is unknown. Interpretations I and II correspond to the dominant effect on F originating at A and D, respectively. The case where the competing influences on F are of similar magnitude and cancel has not been eliminated and will appear as a part of the predictions of both interpretations.

Certain ordinal information on the propagation of the fault can be derived from the simulation tree. For example, in interpretation I it is deduced that deviation of C must precede deviation of D. However, without time delay information, the order of deviation of C and F (and hence D and G) is undeterminable. Table 1 shows the sensor patterns that may arise during fault propagation in interpretations I and II. Note that the case where the effects of A and D on F cancel ($C = D = 1$, $F = G = 0$) is included under both interpretations in Table 1. All process variables not accessible from the root node must be normal (0), under the single fault assumption.

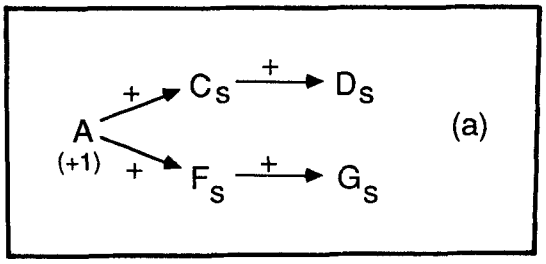
The patterns in Table 1 are the complete set that would cause the diagnostic algorithm of Shiozaki et al. to produce $A(+1)$ as a possible fault origin. Included in these patterns is the true response to a fault entering the plant at A, along with certain spurious patterns. The actual response of the plant corresponds to a specific time-ordered pattern of symptoms (possibly dependent on the fault magnitude). Under the previous assumptions, given n measurements accessible from the root node, only n patterns may arise during the dynamic propagation of the fault. The additional patterns are consistent with the digraph model, but match no real behaviors in the plant.

The information in Table 1 could be used in the diagnosis by matching the stored patterns with on-line sensor data. This is known as the fault dictionary approach (Berenblut and Whitehouse, 1977). Each pattern arising during the propagation of the fault is included in the fault dictionary. In a large plant there will be a large number of patterns associated with each fault, making this approach somewhat unattractive. We will shortly derive a much more concise representation of these data, in the form of a single rule.

Simulation of control loops

Feedback control loops are designed to prevent significant deviation of a controlled variable by transference of disturbances to one or more manipulated variables. In the digraph, disturbances can pass through a controlled variable node without causing significant deviation. This requires special considerations in the synthesis of simulation trees containing control loops.

A single-input/single-output feedback control loop is shown in Figure 5, where A represents a disturbance, B the controlled



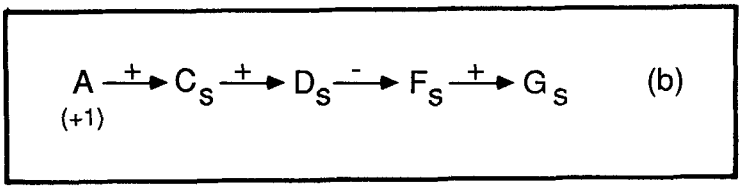


Figure 4. Simulation trees for example 1.
a. Interpretation I
b. Interpretation II

Table 1. Measurement Patterns for Interpretations I and II

Interpretation I				Interpretation II			
C	*D*	*F*	*G*	*C*	*D*	*F*	*G*
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
1	1	0	0	1	1	0	0
0	0	1	0	1	1	−1	0
0	0	1	1	1	1	−1	−1
1	0	1	0				
1	1	1	0				
1	0	1	1				
1	1	1	1				

variable, C the manipulated variable, and D and E downstream variables causally linked to the controlled and manipulated variables., respectively. (More complex controller configurations will be considered subsequently.) A, C, D, and E can be measured or unmeasured. Two interpretations are needed when a disturbance enters this control loop: one corresponding to the case of perfect control of the controlled variable, and one for the case of loop saturation, when the magnitude of the disturbance exceeds the ability of the loop to compensate. In the former case, B (and hence E) will remain normal, and C will respond in the direction compensating for the disturbance, effectively giving the digraph in Figure 6a. In the case of loop saturation or set point changes, the controlled variable will deviate, passing the disturbance to E. The effective digraph for this case is shown in Figure 6b.

The digraphs of Figure 6 apply when the root node is outside the control loop. Failures within a control loop are developed in the manner of example 1, except failure of the controlled variable sensor. In this case, the sign of the controlled variable and its sensor mode will be different, and a separate sensor node is required, Figure 7.

Boolean Representation of the Simulation Tree

The structure of the simulation tree allows it to be converted into a statement of logic. Consider a branch $A \xrightarrow{\pm} B$ in the simulation tree. Since each node in the tree has exactly one input, if $A = +1$, B must be $+1$ or 0 (the latter value accounts for time delay or disturbance damping on the branch). If A is 0, B must be 0. If A is -1, B must be -1 or 0. Table 2 shows the truth table for this relationship, and for $A \xrightarrow{-} B$. It is also useful to define a branch with zero gain, $A \xrightarrow{0} B$, referring to the nonexistence of a branch. This relationship is satisfied as long as $B = 0$. The truth table is shown in Table 2.

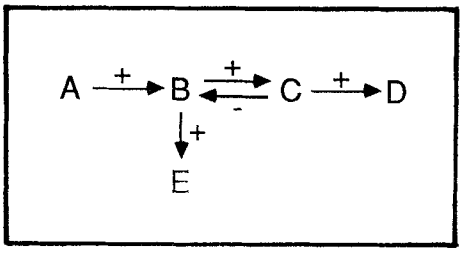


Figure 5. Digraph for control loop.

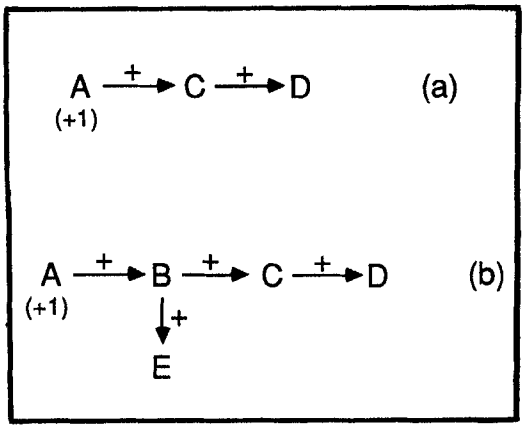


Figure 6. Interpretations of control loop behavior.
a. Loop working
b. Loop saturated

We can define the logical functions p, m, and z as follows:

$$(pAB) \leftrightarrow (A = B) \text{ or } (|A| > |B|)$$
$$(mAB) \leftrightarrow (A = -B) \text{ or } (|A| > |B|)$$
$$(zAB) \leftrightarrow (B = 0) \qquad (1)$$

It is easily shown that the truth tables for p, m, and z are the same as the truth tables for the $+$, $-$, and 0 branches, respectively. These expressions capture the meaning of a simulation tree branch.

In the previous example, the logical relationships implied by the simulation tree in Figure 4a (interpretation I) are:

$$A = +1$$
$$A \xrightarrow{\pm} C \leftrightarrow (pAC)$$
$$C \xrightarrow{\pm} D \leftrightarrow (pCD)$$
$$A \xrightarrow{\pm} F \leftrightarrow (pAF)$$
$$F \xrightarrow{\pm} G \leftrightarrow (pFG) \qquad (2)$$

(The subscripts s have been dropped for notational simplicity). If any of the logical relationships in Eq. 2 are violated by the on-line data, the state of the plant is not consistent with interpretation I. If the state of the plant is inconsistent with both inter-

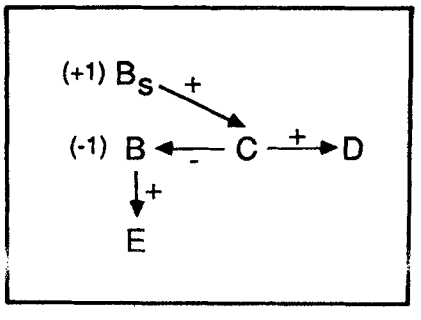


Figure 7. Simulation tree for failure of controlled variable sensor.

Table 2. Truth Tables for Simulation Tree Branches

$A \xrightarrow{+} B$

A \ B	1	0	−1
1	T	T	F
0	F	T	F
−1	F	T	T

$A \rightarrow B$

A \ B	1	0	−1
1	F	T	T
0	F	T	F
−1	T	T	F

$A \xrightarrow{0} B$

A \ B	1	0	−1
1	F	T	F
0	F	T	F
−1	F	T	F

pretations I and II, A cannot be considered a possible fault origin.

The relationships in Eq. 2 are not entirely satisfactory for on-line evaluation since they involve the unmeasured node A. This can be easily remedied by noting that the condition $[(x = +1)$ and $(pxy)]$ is equivalent to $y \neq -1$. Therefore, Eq. 2 can be rewritten in IF-THEN form as:

$$\text{IF } [(C \neq -1) \text{ and } (F \neq -1) \text{ and } (pCD) \text{ and } (pFG)] \text{ THEN } A = +1 \text{ is a possible fault origin} \quad (3)$$

The components of the premise of Eq. (rule) 3, referred to as clauses, are joined with "and" because the conditions implied by all branches must be satisfied by process data for A to be a plausible fault origin, if interpretation I is assumed. This simple rule captures all measurement patterns corresponding to interpretation I in Table 1, and no others. Thus, nine entries in the fault dictionary can be replaced by the single rule above.

Following the same methodology, a rule can be derived for interpretation II:

$$\text{IF } [(C \neq -1) \text{ and } (pCD) \text{ and } (mDF) \text{ and } (pFG)] \text{ THEN } A = +1 \text{ is a possible fault origin} \quad (4)$$

This rule captures the final five entries in Table 1.

Reduction to a single rule

Without quantitative information, it is unknown whether the dominant effect on F is transmitted from A or D. Therefore, any measurement pattern that matches a pattern from Table 1 should trigger $A = +1$ as a possible fault origin. One could conceive of a separate diagnostic rule for each interpretation, as in rules 3 and 4, above. However, this approach may become unwieldy if the number of interpretations is large. It is desirable to represent all interpretations of a given fault by a single rule that captures all measurement patterns associated with the root node.

To produce a single rule covering both interpretations, observe that since rules 3 and 4 share the same conclusion, they can be combined with an "or" operator as follows:

$$\text{IF } \{[(C \neq -1) \text{ and } (F \neq -1) \text{ and } (pCD) \text{ and } (pFG)] \text{ or } [(C \neq -1) \text{ and } (pCD) \text{ and } (mDF) \text{ and } (pFG)]\} \text{ THEN } A = +1 \text{ is a possible fault origin} \quad (5)$$

Using the logical distributive law, $[(x \text{ and } y) \text{ or } (x \text{ and } z)] \leftrightarrow [x \text{ and } (y \text{ or } z)]$, rule 5 can be simplified to:

$$\text{IF } \{(C \neq -1) \text{ and } (pCD) \text{ and } (pFG)] \text{ and } [(F \neq -1) \text{ or } (mDF)]\} \text{ THEN } A = +1 \text{ is a possible fault origin} \quad (6)$$

The premise of this rule is true when any measurement pattern from Table 1 is encountered, and false otherwise. This rule covers both interpretations I and II, and is the only rule needed to diagnose this fault.

Direct Derivation of Rules from the SDG

In general, the procedure of deriving a rule for each interpretation and then combining and simplifying the rules, as suggested in the previous section, is not feasible because a large number of interpretations can be derived from realistic digraphs. The number of interpretations depends on the number of nodes where fault propagation pathways converge, and the number of control loops. Each control loop through which the fault propagates can be either working or saturated. If there are N control loops, there are 2^N possible configurations of working and saturated loops, each giving rise to a separate interpretation. When fault propagation paths converge at a node, the dominant effect can be transmitted from any path entering the node, provided that a connected simulation tree results (see below). For each node where n_i paths converge, the total number of interpretations is multiplied by n_i. Thus, if there are i measured nodes where n_i paths converge, up to $\Pi_{i=2}^{\infty}(n_i)^i$ interpretations can be generated. For example, if there are three nodes where two paths converge, and one node where three paths converge, there may be $2^3 \cdot 3^1 = 24$ separate interpretations, corresponding to different combinations of dominant pathways. These interpretations form a product with the 2^N interpretations arising from control loops, as an upper limit on the total number of interpretations.

In general, converging pathways represent a choice in construction of the simulation tree where one of the paths is invoked to form an interpretation. The full set of interpretations comes from invoking all combinations of these choices. Hence, the combined set of interpretations can be represented by making explicit the choices, instead of enumerating each interpretation. Again, it should be noted that although the interpretations relate to discrete choices of the dominant causal pathways, the interpretations include all possible behaviors of the real system.

Now consider the partially developed simulation tree shown in Figure 8 (example 2), assuming all nodes are measured. There are two nodes (D and F) where two feedforward paths

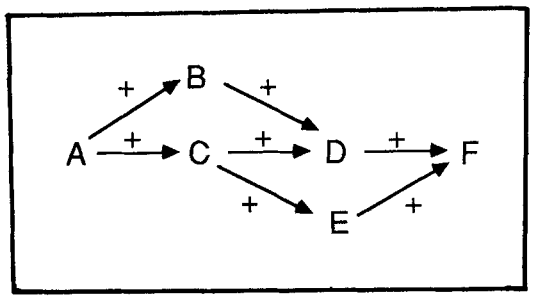


Figure 8. Digraph for example 2.

converge, resulting in four interpretations, shown in Figure 9a–d. The combined set of interpretations can be described by the following set of branches:

$$A \xrightarrow{\pm} B \text{ and } A \xrightarrow{\pm} C \text{ and } C \xrightarrow{\pm} E$$
$$\text{and } (B \xrightarrow{\pm} D \text{ eor } C \xrightarrow{\pm} D)$$
$$\text{and } (D \xrightarrow{\pm} F \text{ eor } E \xrightarrow{\pm} F), \tag{7}$$

where "eor" is the "exclusive or," meaning that exactly one branch is selected. Using the logical operators defined in Eq. 1, a rule corresponding to statement 7 can be written:

$$\begin{aligned}&\text{IF } \{(pAB) \text{ and } (pAC) \text{ and } (pCE)\\ &\quad \text{and } [(pBD) \text{ or } (pCD)]\\ &\quad \text{and } [(pDF) \text{ or } (pEF)]\}\\ &\text{THEN } A = +1 \text{ is a possible fault origin}\end{aligned} \tag{8}$$

This rule covers all measurement patterns associated with all

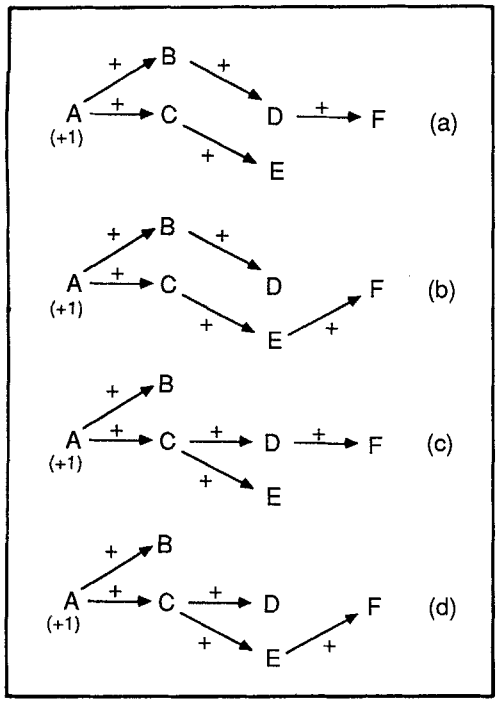


Figure 9. Four interpretations of fault propagation in example 2.

interpretations of the digraph in Figure 8. The exclusive "or" is not used because this would eliminate certain valid measurement patterns. For example, if $D = E = F = 1$, rule 8 would be false if "eor" were used, although this pattern is part of several valid interpretations. Use of the inclusive "or" leads to the correct set of measurement patterns.

In general, the diagnostic rule is derived from the SDG by adding a clause of the following form for each measured node n_i:

$$\ldots \text{and } [(*k_1 n_i) \text{ or } (*k_2 n_i) \ldots \text{or } (*k_J n_i)], \tag{9}$$

where k_j ($j = 1, \ldots, J$) are the inputs to n_i from the SDG, and * is m, p, or z, corresponding to the sign of the branch from k_j to n_i. The signs can be conditional on the state of the process, as explained below.

Positive feedback loops

When feedback loops are present, certain choices of fault propagation pathways will result in isolation of a feedback loop from the rest of the digraph. If isolation of a loop occurs, spurious measurement patterns can result. For example, consider Figure 10 (example 3), with $A = +1$ as the fault origin. Utilizing clause 9, the rule for this digraph would be:

$$\begin{aligned}&\text{IF } \{[(pAB) \text{ or } (pCB)]\\ &\quad \text{and } [(pAC) \text{ or } (pBC)]\}\\ &\text{THEN } A = +1 \text{ is a possible fault origin}\end{aligned} \tag{10}$$

This rule corresponds to four interpretations, whereas only three interpretations are valid, namely: $A \xrightarrow{\pm} B \xrightarrow{\pm} C$, $A \xrightarrow{\pm} B$ and $A \xrightarrow{\pm} C$, and $A \xrightarrow{\pm} C \xrightarrow{\pm} B$. By including the interpretation $B \xrightarrow{\pm} C$ and $C \xrightarrow{\pm} B$, rule 10 admits the measurement pattern $B = C = -1$, which is inconsistent with the fault origin $A = +1$. To remove this spurious pattern, observe that at least one branch must enter the feedback loop to avoid isolation of B and C from the root node. The condition:

$$[(pAB) \text{ or } (pAC)] \tag{11}$$

joined with "and" to the premise in rule 10, forces one input to the feedback loop and limits the acceptable measurement patterns to those associated with the three valid interpretations. In general, each positive feedback loop with input branches

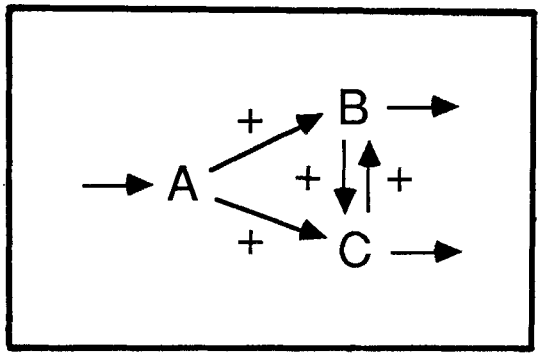


Figure 10. Digraph for example 3.

$B_i (i = 1, \ldots, I)$ requires an extra clause in the form:

$$\ldots \text{and } (\underline{B_1} \text{ or } \underline{B_2} \ldots \text{or } \underline{B_I}), \tag{12}$$

where $\underline{B_i}$ is the m, p, or z function for the branch B_i.

Introduction of a clause like 12 is required only for positive feedback loops (where the product of branch signs around the loop is positive). Isolated negative feedback loops have no consistent nonzero solution, so no spurious measurement patterns are associated with loop isolation. Trivial feedback loops are those with only one input; in these cases, the feedback path can always be ignored, since the requirement of one input to the loop eliminates the feedback path from all valid interpretations.

Removal of unmeasured nodes

Unmeasured nodes are removed from the digraph after selection of a root node. As seen in example 1, nodes are removed by formation of composite branches from the branches that pass through unmeasured nodes. The sign of a composite branch is the product of the signs of the branches it replaces. When eliminating unmeasured nodes in this fashion, two undesirable side effects can occur, namely, the loss of a signed constraint between nodes, and the introduction of a spurious pathway. These side effects are infrequent and can occur only when there are multiple pathways from the root node to an unmeasured node.

Consider Figure 11a, in which two paths of opposite sign from the root node terminate at A. Removal of the unmeasured node A by forming composite branches is shown in Figure 11b. The restriction imposed by the original digraph that B and C deviate in the same direction is lost when A is removed. Therefore a rule derived directly from Figure 11b would include spurious interpretations, and the extra clause:

$$\text{and } [(pBC) \text{ or } (pCB)] \tag{13}$$

which yields false if B and C deviate in opposite directions, is required when A is removed.

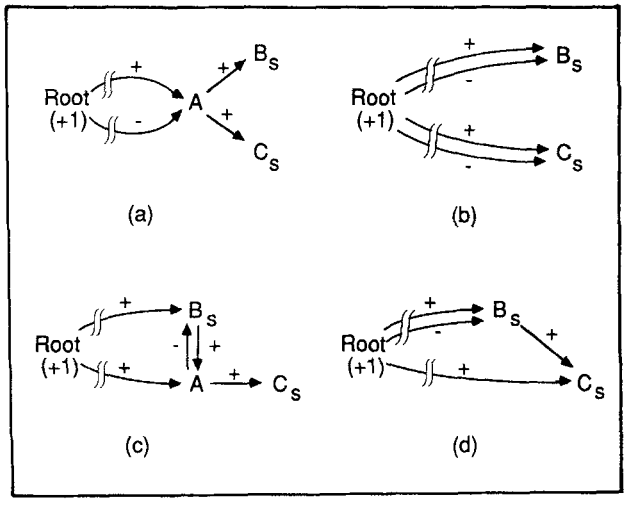


Figure 11. Removal of unmeasured nodes requiring extra clauses.
a., c. Full digraphs
b., d. Reduced digraphs

In Figures 11c and 11d a similar situation is encountered. When A is removed, C can assume the value -1, which is not possible in the original digraph. The pathway $Root \xrightarrow{} B \xrightarrow{\pm} C$ is spurious, since it requires A to deviate simultaneously in two directions. To avoid spurious interpretations, the clause:

$$\text{and } (C \neq -1) \tag{14}$$

is added to the rule when A is removed, for a $+1$ deviation of the root node.

Conditional branches

The existence or sign of certain branches of the digraph may be dependent on the process state. For example, pressure or flow transmission cannot occur through a closed valve. An example of a branch whose sign is dependent on process temperatures is given in Shiozaki et al. (1985, p. 293). Conditional signs can be included in the diagnostic rules without modification of the basic rule syntax. A branch with conditional sign can be expressed as:

$$(xAB) \text{ where } x = \begin{cases} p \text{ if cond}_1 \\ m \text{ if cond}_2 \\ \text{else } z \end{cases} \tag{15}$$

This formula is easily implemented in LISP, since the result of a LISP conditional can be a function. Alternatively, statement 15 can be expressed in terms of fixed functions as:

$$\{[(pAB) \text{ and cond}_1] \text{ or } [(mAB) \text{ and cond}_2] \text{ or } (zAB)\} \tag{16}$$

which follows from statement 15 since $[(*AB) \text{ or } (zAB)] \leftrightarrow (*AB)$, where $*$ is m or p.

For example, the conditional absence of a branch $A \xrightarrow{+} B$, which appears only if a valve is open, is represented as:

$$\begin{aligned} &(xAB) \text{ where } x = p \text{ if valve open, else } x = z \\ &\leftrightarrow \{[(pAB) \text{ and (valve open)}] \text{ or } (zAB)\} \end{aligned} \tag{17}$$

This expression requires information on valve positions. If the required valve positions are not available as on-line measurements, the information must be entered manually.

Control loops

We have shown previously that two interpretations are necessary to describe the possible behaviors of a single-input/single-output (SISO) control loop. A single clause covering both normal and saturated behavior can be derived by joining the logical conditions for each state with "or." From Figure 6, the following describes the possible behaviors:

$$\begin{aligned} &(B = 0) \text{ and } (pAC) \text{ and } (pCD) \text{ and } (E = 0) \\ &\text{or } [(pAB) \text{ and } (pBC) \text{ and } (pCD) \text{ and } (pBE)] \end{aligned} \tag{18}$$

This expression is true whenever a disturbance is passed into the control loop through its controlled variable, whether or not the loop saturates.

It can be shown that expression 18 is equivalent to the following:

$$[(pAB) \text{ or } (mCB)]$$
$$\text{and } [(xAC) \text{ or } (pBC)]$$
$$\text{and } (pCD)$$
$$\text{and } (pBE) \qquad (19)$$

where $x = p$ if $B = 0$, and $x = z$ if $B \neq 0$. The advantage of this expression is that it can be derived directly from Figure 5 using clause 9, except for the clause concerning node C.

The difference is rationalized as follows. All branches in Figure 5 are conventional branches except the branch $B \xrightarrow{+} C$, which does not obey the normal branch definition of Table 2, since for "working" behavior of the loop, Figure 6a, B is normal while C is disturbed. This results from integral action or high gain causing zero offset in B. In effect, disturbances can be transmitted to C from two sources, corresponding to Figures 6a and 6b. If B is normal, then the disturbance is transmitted from A; otherwise, it is transmitted from B to C.

In general, SISO and more complex control loops can be handled by the introduction of nonphysical, conditional branches between each disturbance entering the control loop and the manipulated variable(s) of the loop. For an $n \times n$ controller ($n \geq 1$), with d external disturbances, $n \times d$ of these branches are required. After these branches are introduced, the normal rule development using clause 9 applies.

Summary of rule construction procedure

The procedure for generating rules from a digraph is as follows:

1. Select a root node. The conclusion of the rule developed in the subsequent steps is the list of faults corresponding to the root node.
2. Remove branches that are input to the root node.
3. Lump sensor nodes for sensors that are not malfunctioning, if the measurement is rapid compared to fault propagation.
4. Remove inaccessible nodes from the digraph (nodes that cannot be reached from the root node). For each measured inaccessible node, add a clause to the diagnostic rule setting the node to zero.
5. Remove unmeasured nodes from the digraph, forming branches between measured nodes only. If the root node and an unmeasured node are connected by multiple pathways, clauses analogous to 13 and 14 may be required. Trivial feedback loops can be removed at this stage.
6. Add conditional branches connecting disturbances and manipulated variables in functioning control loops.
7. For each measured node, add a clause in the form of clause 9.
8. For each positive feedback loop, add a clause in the form of clause 12.
9. (optional) Simplify the rule using rules of logic. The absorption rule [x and (x or y)] $\leftrightarrow x$ will often allow deletion of certain clauses.

Application example

To demonstrate the direct derivation of diagnostic rules, we consider the recycle reactor system shown in Figure 12. In this example, an exothermic positive-order reaction of A to B takes place in a stirred-tank reactor. To provide temperature control, part of the reactor outlet stream is recycled to the reactor through a heat exchanger. The recycle flow rate is controlled, and the reactor residence time is controlled by maintaining a constant level in the reactor. Constant boundary pressures and constant physical properties are assumed. The SDG for this system is shown in Figure 13, derived from the system equations by the method of Iri et al. (1979).

The root node chosen for purposes of example is $C_L = +1$, failure of the level controller with positive deviation of the con-

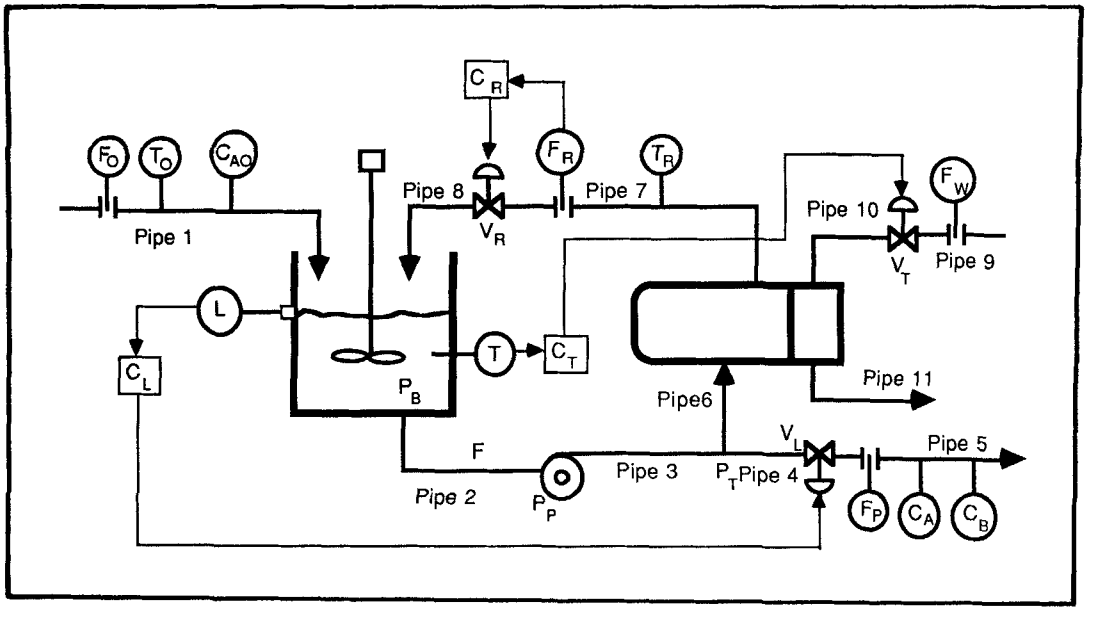


Figure 12. Flowsheet of stirred-tank reactor with recycle.

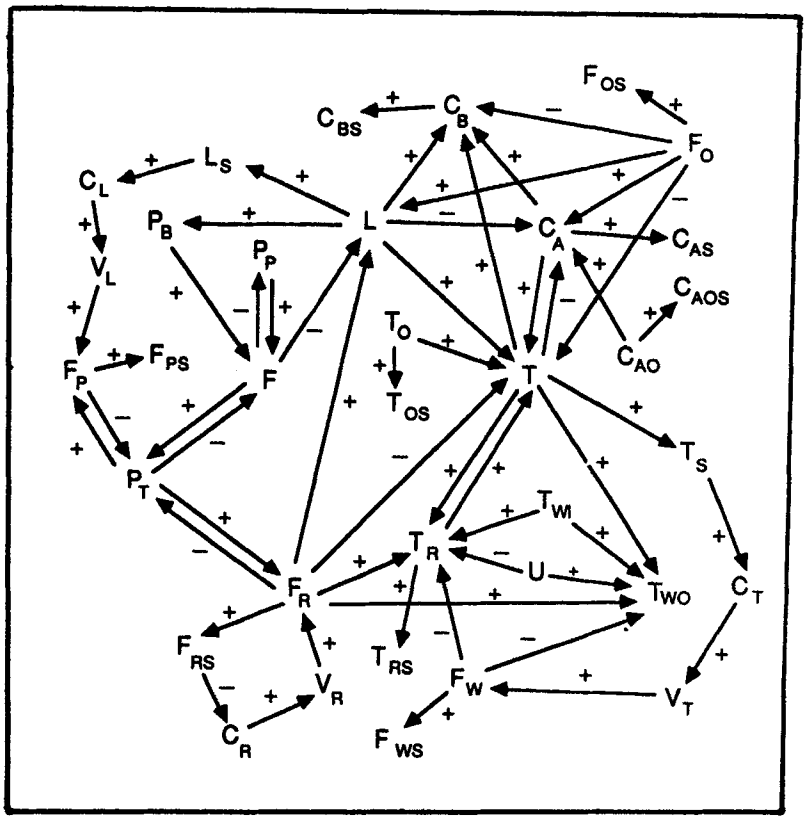


Figure 13. Digraph of stirred-tank reactor system.

troller output. Steps 1 through 6 in derivation of the diagnostic rule for this root node result in the digraph shown in Figure 14 (subscripts s have been dropped for simplicity). Five conditional branches for the reactor temperature and recycle flow rate control loops are shown. No clauses are required in this example when unmeasured nodes are removed.

The inaccessible measured nodes for this root node are F_O, C_{AO}, and T_O. The first clause in the diagnostic rule is therefore:

$$(F_O = 0) \text{ and } (C_{AO} = 0) \text{ and } (T_O = 0) \qquad (20)$$

Step 7 of the method yields one clause for each accessible, measured node:

$$\begin{aligned}
&\text{and } (pC_LF_P)\\
&\text{and } [(mF_PF_R) \text{ or } (pC_RF_R)]\\
&\text{and } [(mF_PL) \text{ or } (pF_RL)]\\
&\text{and } [(pLC_B) \text{ or } (pC_AC_B) \text{ or } (pTC_B)]\\
&\text{and } (pC_TF_W)\\
&\text{and } [(pF_RT_R) \text{ or } (pTT_R) \text{ or } (mF_WT_R)]\\
&\text{and } [(pLT) \text{ or } (pC_AT) \text{ or } (mF_RT) \text{ or } (pT_RT)]\\
&\text{and } [(mLC_A) \text{ or } (mTC_A)]\\
&\text{and } [(x_1LC_T) \text{ or } (x_2F_RC_T) \text{ or } (x_3F_RC_T) \text{ or}\\
&\quad (x_4C_AC_T) \text{ or } (pTC_T)]\\
&\text{and } [(x_5F_PC_R) \text{ or } (mF_RC_R)] \qquad (21)
\end{aligned}$$

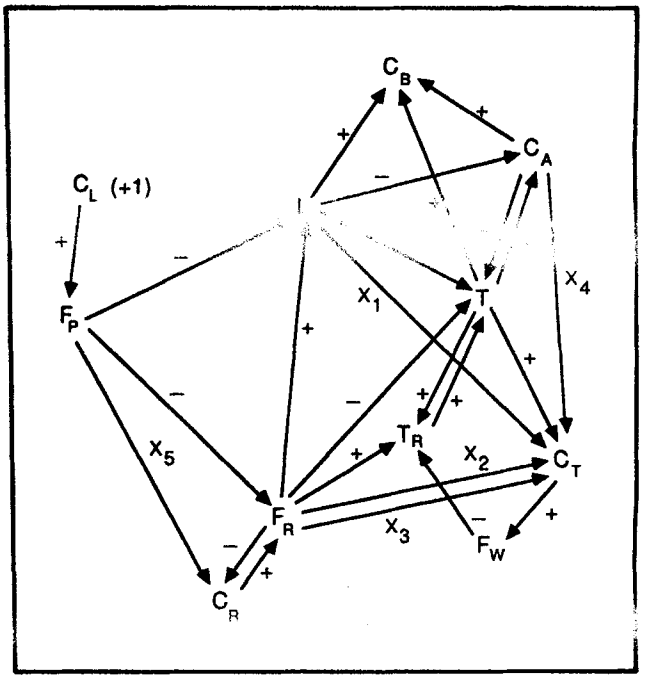


Figure 14. Reduced digraph of stirred tank reactor system, $C_L = +1$ as root node.

where

$$x_1, x_4 = p \text{ if } T = 0, \text{ else } z$$

$$x_2 = m \text{ if } T = 0, \text{ else } z$$

$$x_3 = p \text{ if } T = 0 \text{ and } T_R = 0, \text{ else } z$$

$$x_5 = p \text{ if } F_R = 0, \text{ else } z$$

There is one positive feedback loop in the simplified digraph, involving T and T_R. This loop requires a clause assuring at least one input into the loop, from step 8 of the algorithm:

$$\text{and } [(pLT) \text{ or } (pC_AT) \text{ or } (mF_RT) \text{ or } (pF_RT_R) \text{ or } (mF_WT_R)] \quad (22)$$

Together, statements 20–22 form the entire premise of a rule concluding possible level controller fails high.

Computational Considerations

Preparation of the diagnostic rules is done off-line. For each node of the digraph, two rules are required, corresponding to positive and negative perturbations of the node. Rule generation can be automated to reduce manual effort. Effort in preparing the rules is not critical, and depends on programming details and the computing environment. Computation speed is much more important in the on-line environment where rule evaluation occurs. Here, timely response by the operator is dependent on the speed of the diagnosis, and if the computation time is too long, the diagnostis can lose its utility. The current method involves only simple arithmetical operations on integers with no iteration or recursion, which can be executed with extreme rapidity in the on-line environment.

To establish the speed of the method, we determine the computational time, not including data acquisition, used by the current method on the example of Shiozaki et al. (1985). This example involves 207 branches, 36 measured nodes, and 99 total nodes. Computation times reported for the algorithmic technique ranged up to 5 min on a FACOM M-200 computer. In the rule-based approach, 198 rules with approximately 36 clauses per rule are required. Using FORTRAN on a Data General MV-4000, evaluation of these rules required 0.02 s. Less than 2 s were required with an IBM PC using BASIC.

Discussion

We have shown in this paper how the digraph can be converted to a set of diagnostic rules that can be used in an on-line environment for fault diagnosis. It has been pointed out that the rule-based approach to fault diagnosis is computationally more efficient than the previous algorithmic approach. The rule-based format facilitates a number of additional improvements to the digraph methodology, which are summarized below.

1. *Diagnostic Resolution.* One of the drawbacks of digraph-based methods is that the resolution of the diagnosis may be poor. In the pilot plant study of Tsuge et al. (1985), the SDG algorithm using on-line sensor data yielded an average of 23 fault candidates out of 53 possibilities. The current method suggests a means for improving the diagnostic resolution, without requiring more measurements or more complex representations, through reduction of the number of spurious interpretations that are included in the diagnostic rule. Spurious interpretations are generated because of the ambiguity of the SDG in predicting the fault propagation pathways. Knowledge of the actual pathways of fault propagation can be incorporated into the diagnostic rule. Such knowledge can be derived from numerical simulation, qualitative simulation (Bobrow, 1985; Oyeleye and Kramer, 1986), or from operating experience.

To demonstrate, numerical simulation (with a specific set of numerical parameters) was used to reduce the set of interpretations for the application example. The simulations provided the direction of change of the measured variables and the order of events for several different magnitudes of the fault $C_L = +1$. This knowledge was used to identify the dominant pathways in the digraph. As a result, the number of interpretations was reduced to four, corresponding to normal/saturated control loop behaviors that are dependent on the fault magnitude, resulting in the following rule:

$$\begin{aligned}\text{IF } \{&(F_O = 0) \text{ and } (C_{AO} = 0) \text{ and } (T_O = 0)\\ &\text{and } (pC_LF_P) \text{ and } (mF_PF_R)\\ &\text{and } [(x_5F_PC_R) \text{ or } (mF_RC_R)]\\ &\text{and } (mF_PL) \text{ and } (pLC_B)\\ &\text{and } [(x_1LC_T) \text{ or } (pTC_T)]\\ &\text{and } (pC_TF_W) \text{ and } (mF_WT_R) \text{ and } (pLT)\\ &\text{and } (mLC_A)\}\\ \text{THEN possible-level-controller-fails-high} \quad &(23)\end{aligned}$$

The original rule, statements 20–22, yields true for 647 measurement patterns. In rule 23, only 133 patterns satisfy the rule, reducing by a factor of five the number of patterns triggering this diagnosis. In general, the less ambiguous the model of the fault used to construct the rules, the fewer spurious diagnoses will be generated.

2. *Specialized Branches.* The + and − branches used in this work have the particular meanings defined by the truth tables shown in Table 2. Although most general, these are not the only types of interactions that can be defined. For example, suppose B is an optimized quantity that increases whenever A is perturbed, regardless of the direction of perturbation of A. The truth table for this interaction is given in Table 3. A logical function analogous to p or m can easily be constructed from this truth table. Whenever an optimized quantity appears in the digraph, this function would be inserted, with no change in the overall rule syntax. In an algorithmic context, this simple modification would require significant revision of the code.

Another type of specialized branch is a branch with time delay. Time delays would help distinguish sensor failures from

Table 3. Truth Tables for Optimized Quantities

A \ B	$\|A\| \xrightarrow{+} \|B\|$ 1	0	−1	$\|A\| \xrightarrow{-} \|B\|$ 1	0	−1
1	T	T	F	F	T	T
0	F	T	F	F	T	F
−1	T	T	F	F	T	T

other process failures, and could improve resolution for other types of malfunctions as well. We anticipate that by defining the proper logical relation in place of p and m, branches with time delay could be included without modification of the basic rule syntax.

3. *Integration with Other Knowledge.* Expression of the diagnostic criteria in the form of rules at once opens all the potentialities of the expert systems framework, which is an ideal format for integrating knowledge from disparate sources (Kramer, 1986b). For example, forward chaining could be used to trigger other diagnostic rules to help refine or increase the certainty of the fault hypothesis, as suggested in the following:

IF (digraph rule premise)
THEN possible-reactor-leak

IF possible-reactor-leak and mass-balance-violation
THEN likely-reactor-leak

IF likely-reactor-leak and H_2S-smell
THEN definite-reactor-leak

The additional diagnostic rules may be heuristics, or based on quantitative models, as in the method of governing equations (Kramer, 1986a). The Boolean values of mass-balance-violation and H_2S-smell would be determined by backward chaining (possibly ending in queries to the operator).

A natural extension of any diagnostic system is to incorporate rules on corrective action. For example, the rules above could trigger a rule advising the operator:

IF likely-reactor-leak
THEN message "Execute reactor shut down procedure"

Other rules concerning process operation or optimization could similarly be included in the expert system. Additional considerations in real-time expert systems have been discussed by Moore et al. (1985).

Acknowledgment

The authors thank the Shell Foundation and the Atlantic Richfield Foundation for support of this work. The authors are grateful to O. O. Oyeleye, who prepared the digraph for the application example. A portion of this work was carried out under National Science Foundation Grant No. CBT-8605253.

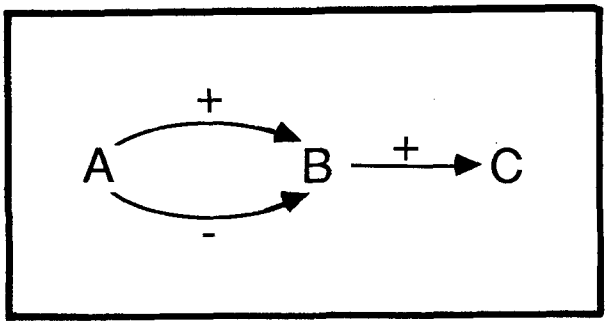


Figure A1. Limitation to single transition.

Table A1. Possible Measurement Patterns for Figure A1 Digraph

A	B	C
1	0	0
1	1	0
1	1	1
1	0	1*
1	−1	1*
1	−1	0
1	−1	−1

*Inconsistent patterns, see Appendix text

Notation

A, B, C, etc. = digraph nodes or associated qualitative state
C_A, C_B, C_{AO} = concentrations of A and B in reactor, A in feed
C_L, C_R, C_T = controller output signals, level, recycle flow, and temperature
F, F_O, F_P, F_R, F_W = reactor outlet, feed, product, recycle, cooling water flow rates
L = liquid level in reactor
m = logical condition for negative branch
p = logical condition for positive branch
P_B, P_T = tank outlet, recycle takeoff pressures
P_P = pump head
S = sensor reading
$T, T_O, T_R, T_{WI}, T_{WO}$ = reactor, feed, recycle stream, cooling water inlet, outlet temperatures
U = heat transfer coefficient
V_L, V_R, V_T = valve stem positions; level, recycle, temperature loops
z = logical condition for zero gain branch

Appendix: Limitation to Single Transitions

Consider the digraph of Figure A1, with $A = +1$ as the fault origin. Suppose that initially the dominant branch of the digraph is the positive branch from A to B, and subsequently is the negative branch. A set of measurement patterns possibly generated in this situation are listed in Table A1. Two of the patterns, indicated by asterisks, are inconsistent with the usual meaning of a digraph branch. In these cases, the method of Shiozaki et al. (1985) will exclude the actual fault. These inconsistencies occur due to time delay between variables when a variable undergoes a reversal of direction. If only single changes of state are allowed, this type of inconsistency is avoided.

Literature cited

Berenblut, B. J., and H. B. Whitehouse, "A Method for Monitoring Process Plant Based on a Decision Table Analysis," *Chem. Eng. (London)*, **318**, 175 (1977).

Bobrow, D. G. ed., *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, MA (1985).

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

Iri, M., K. Aoki, E. O'Shima, and H. Matsuyama, "An Algorithm for Diagnosis of System Failures in the Chemical Process," *Comput. Chem. Eng.*, **3**, 489 (1979).

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

Kokawa, M., S. Miyazaki, and S. Shingai, "Fault Location Using Digraph and Inverse Direction Search with Application," *Automatica*, **19**, 729 (1983).

Kramer, M. A., "Malfunction Diagnosis Using Quantitative Models

and Non-Boolean Reasoning in Expert Systems," *AIChE J.*, **33**, 130 (1987).

———, "Integration of Heuristic and Model-Based Inference in Chemical Process Fault Diagnosis," *IFAC Workshop Fault Detection, Safety in Chem. Plants*, Kyoto (1986b).

Lees, F. P., "Process Computer Alarm and Disturbance Analysis: Review of the State of the Art," *Comput. Chem. Eng.*, **7**, 669 (1983).

Moore, R. L., L. B. Hawkinson, M. E. Levin, and C. G. Knickerbocker, "Expert Control," *Proc. ACC*, Boston, 885 (1985).

Oyeleye, O. O., and M. A. Kramer, "Qualitative Simulation of Chemical Process Systems: Steady State Analysis," *Chem. Eng. Sci.*, submitted (1987).

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

Shiozaki, J., H. Matsuyama, E. O'Shima, and M. Iri, "An Improved Algorithm for Diagnosis of System Failures in the Chemical Process," *Comput. Chem. Eng.*, **9**, 285 (1985).

Tsuge, Y., J. Shiozaki, H. Matsuyama, and E. O'Shima, "Fault Diagnosis Algorithms Based on the Signed Directed Graph and its Modifications," *Ind. Chem. Eng. Symp. Ser.*, **92**, 133 (1985).

Umeda, T., T. Kuryama, E. O'Shima, and H. Matsuyama, "A Graphical Approach to Cause and Effect Analysis of Chemical Processing Systems," *Chem. Eng. Sci.*, **35**, 2379 (1980).

Manuscript received July 22, 1986, and revision received Dec. 10, 1986.