FAULT DIAGNOSIS AND COMPUTER-AIDED DIAGNOSTIC ADVISORS

Mark A. Kramer
Gensym Corporation
Cambridge, MA 02140

Roar Fjellheim
Computas Expert Systems A.S.
Sandvika, Norway

Abstract

Virtually all modern process plants include computerized information systems that centralize, present, and alarm sensor data. Yet there remains tremendous scope for improving operator support by adopting computer-based fault detection, identification and supervision systems to satisfy incentives for human safety, environmental safeguards, equipment protection and product quality. This paper reviews requirements, recent progress and remaining challenges in detecting, identifying and correcting faults in process plants, and samples a number of architectures, tools, and industrial applications.

Keywords

Fault diagnosis, Fault detection, Fault identification, Pattern recognition, Neural networks, Inductive learning, Expert systems, Operator support systems.

Introduction

Every industrial process has the potential of deviating outside its normal and intended range of behavior. Unless contained, process deviations may have a serious impact on process economy, safety, product quality and pollution level. Proper mechanisms for preventing, detecting, diagnosing and correcting abnormal process behavior should therefore be an important part of the supervisory control system of any plant. While this is widely recognized, in practice the diagnosis and response tasks are too often characterized by manual, ill-documented or *ad hoc* operator procedures. There is tremendous scope for improvement by adopting computer-based fault diagnosis and advisory systems. The direct benefits to be gained include:

- Increased safety and reduced costs by vigilant monitoring of multiple safety and economic parameters;
- Decreases in emissions, material and energy waste associated with excursions from normal operation;
- Increased product quality by rapid detection and correction of incipient disturbances;
- Reductions in human error due to mis-assessment of the process condition or failure to follow standard procedures;
- Increased plant lifetime by reduction of the duration and severity of out-of-control episodes.

These incentives have stimulated a large number of academic and industrial activities over the last decade. Perusal of the literature reveals a broad, almost bewildering array of proposed diagnostic approaches, spawned from artificial intelligence (expert systems, neural networks, qualitative simulation, case-based reasoning), probability and statistics (statistical process control, chemometrics, Bayesian networks), systems theory (estimators, observers, analytical redundancy), and safety and reliability (fault trees, causal reasoning). A growing number of industrial systems are also represented, each displaying a unique architecture, knowledge representation, solution algorithm, and human-machine interaction. Meanwhile, the process industries have not widely adopted any diagnostic technique (with the possible exception of SPC for fault detection), and there are few standard vendor-supported tools available to support off-the-shelf solutions. The prospects for this dynamic field are indeed difficult to discern.

In this paper, we examine the challenges underlying the design of diagnostic systems, beginning with a perspective on the role and functions of diagnosis systems, moving to design considerations, followed by a review of diagnostic methodologies, and concluding with a sampling of architectures, tools and environments, industrial applications, and future challenges.

Role of Monitoring and Diagnostic Systems

Diagnostic systems fit into the hierarchy of plant management at the execution supervision layer, above regulatory control layer, and below the process planning layer (Fig. 1). The general goal at this level is to assure the success of the planned operations by monitoring the performance of the system and its regulatory controls. At minimum this implies keeping operators, managers, and maintenance personnel better informed about what is going on in the process. The design and implementation of computerized fault diagnosis advisors must be driven by the needs of these user groups.

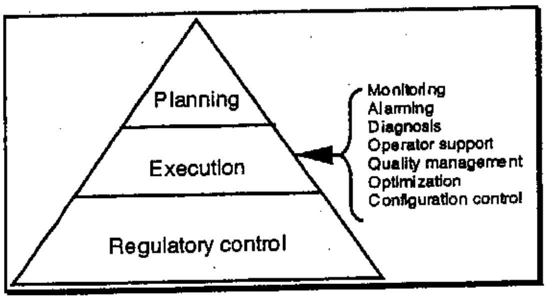

Figure 1. Plant management hierarchy.

The functions and information flow in an on-line diagnostic system are shown in Fig. 2. The diagram assumes a process monitored and controlled by a SCADA system, and supervised by an operator or operational team. Because many factors can influence the precise requirements, diagnostic applications may involve only selected elements of these functionalities, and additional non-diagnostic functions may play an important role. Nor do these functions necessarily occur sequentially in the order shown; for example, isolation and identification are not necessarily required for prognosis and compensation. The following functions may be included:

- Feature Extraction. The diagnostic process starts from process data and alarms supplied by a SCADA system, manual inputs from the operator, and information from the plant database (such as laboratory results). Normally it is desirable to process this data to extract features relevant to the fault detection and diagnosis tasks, rather than relying on raw data alone.
- *Prediction*: Ideally, one would like to prevent faults before they occur. This implies a predictive ability that might be realized through monitoring cumulative wear and shocks, composition analysis of lubrication fluids, extrapolation of real-time trends, etc.
- *Detection:* Detecting that a fault has actually occurred is not always obvious, due to slow induction or compensation by the control system. Early detection may provide invaluable warning on emerging problems, thus enabling the operator to issue actions that avoid serious process upsets.
- *Isolation:* In order to handle the fault, it is necessary to narrow the location of origin. If the physical unit or functional subsystem that is the origin of the fault can be isolated, this may be sufficient for error handling.
- *Identification:* This step involves determining the identity of the fault, usually from a pre-enumerated set of possibilities. Identification may imply one or more of the following:
  (a) *Classification*: Determining the type of fault (e.g. a leak), without necessarily providing other details (extent, time of occurrence, precise location, root cause, etc.).
  (b) *Estimation*: Determining the extent of the fault and other parameter values quantifying the fault.
  (c) *Diagnosis*: Determining the underlying cause for the fault (e.g. corrosion as the cause of a leak).
- *Prognosis:* When an abnormal event is in progress, we would like to know the potential outcome of the event and the time window available for affecting that outcome.
- *Compensation*: In many cases, the first-level response to a fault is to mitigate its negative consequence, for example, by starting a back-up unit.
- *Correction*: Finally, the system may be corrected by repairing or replacing the faulted component or ingredient. The fault handling functions may issue advice to the operator, or take direct actions via the SCADA system.

This diversity of functions implies that there cannot be an all-encompassing approach to diagnosis. As is clear from the literature on the subject, the contributions of AI, statistics and systems theory all play important roles in solving different aspects of the problem. While individual elements of the problem may be fairly well understood, integrating these elements into an overall operating system presents significant engineering challenges.

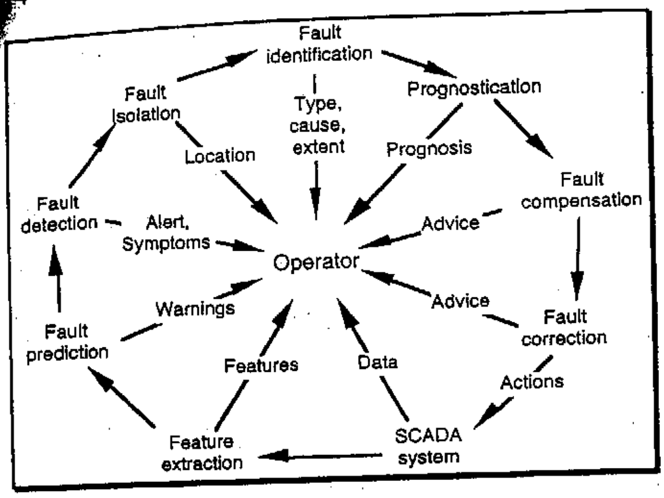

Figure 2. Functions and information flow in diagnostic systems.

Considerations in Diagnostic System Design

In this section, we discuss requirements and raise issues to be considered in the design of a diagnostic system. Each diagnostic function in Fig. 2 is characterized by information flows to the operator. Early in the design and planning of a diagnostic system, there are several strategic decisions that must be made regarding these information flows. These include:

- Which diagnostic functions should be performed autonomously by the system, which on an on-demand basis, and which should be left to the users?
- Should the system or the user initiate and guide the diagnostic reasoning process? Under what circumstances should manual input be depended upon?
- What information should the system display on a continuous basis, and what information should be accessible on demand?
- What strategy should be used to prioritize and control the flow of information from the system to the user?
- How should status information be presented so the user always knows the status of the system at a glance? How should the system focus user attention?
- How should the system alert the operators when a deviation is detected, for example, via text, color coding, sound, animation, or a combination thereof?
- What should the end-user screens look like with respect to color, layout, information density, etc.? How should the user navigate between the screens? What measures should be taken to make sure important information is not missed if the user is "on the wrong page?"

Design decisions concerning the technical scope and approach to the diagnostic system include (see Stephanopoulos and Han, 1994):

- The sources of faults to be considered, including sensors, actuators, controllers, process equipment, process parameters, raw material properties, and mis-operation;
- The failure modes for each fault source, including type of fault, extent, and temporal characteristics (e.g. step versus drift);
- The decomposition boundaries for applying monitoring models and algorithms;
- The representation of normal process behavior used for detecting faults (steady state, dynamic, linear, nonlinear, deterministic, stochastic, etc.);
- The sources of data to consider, including on-line measurements and possibly off-line manual measurements and laboratory analyses;
- Features to be extracted and examined in the data (signal properties, equation residuals, parameter estimates, dimensional projections, symbolic or Boolean discretizations, etc.);
- The type of fault detection tests to be applied (statistical, logical, univariate or multivariate);
- The description of fault behavior used to determine fault identity (algebraic or differential equations, order-of-magnitude relationships, causal graphs, probabilistic models, nonlinear mappings, rules, etc.);
- The algorithm for applying fault description to determine the fault identity.

Of course, the technical approach cannot be considered independent of the type of knowledge that is available to support it, or that its creators are willing to invest the time and effort to create. Possible knowledge sources for developing the diagnostic system include:

- Process and instrumentation diagrams;
- Equipment and material specifications;
- Operational specifications, including standard operating plans, alarm limits, and the like;
- Design and operability analyses such as fault trees, HAZOP studies, etc.;
- Analytical models, for either normal and/or faulty operations;
- Historical data, possibly including both normal and abnormal data;
- Knowledge of operating personnel.

The quality and extent of these knowledge resources will guide the technical approach and may critically impede or facilitate the development and deployment of the diagnostic system.

From an systems integration and software engineering viewpoints, additional requirements may include:

- Interfaces with the plant information systems for accessing real-time, historical and off-line laboratory data, and existing reporting and documentation systems;
- Utilization of a common database of information on plant layout, standard operating procedures, operating ranges, hazard and operability study results, etc.;
- Recording and archiving of data, conclusions, and user inputs;
- Knowledge representation in a form that is transparent, verifiable, and easy to maintain;
- Use of algorithms that are scaleable in terms of computer power, memory, modeling effort, etc., and capable of operating on the same time scale of the disturbance.

Fault Detection, Isolation and Identification

Of the monitoring and diagnosis functions defined above, the core problems of fault detection and identification (FDI) have received the most research attention. Associated problems such as prioritization of symptomatic information, fault prediction, explanation, and response are generally regarded as highly process-specific and resistant to generalized formalization and solution, although there seems to be no fundamental justification for this perception. In this section, diagnosis is described as a three-stage process involving fault detection, isolation and identification. We first discuss these problems in general terms, and then focus on five specific classes of techniques. Additional reviews of FDI are provided by Isermann (1993), Frank (1992), Kim (1994), Kramer and Mah (1993) and Stephanopoulos and Han (1994).

Fault Detection

Fault detection is a model-based task that involves comparison of the observed behavior of the process to a reference model representing fault-free behavior, and detecting significant discrepancies. Fault detection methods are defined by the model used to represent normal behavior, and the nature of the test used to detect deviation from the normal model. No information concerning failure modes and effects is required in the fault detection step.

A general representation of the fault detection problem is shown in Fig. 3. The axes represent features used to detect faults, z_i, which are generally operating conditions and product quality measurements, but may also involve derived quantities such as estimated values of parameters and states. Additionally, features can include operating history through the use of delayed measurements, trends, and explicit elapsed times (e.g. the time from the beginning of a batch). In this space, two types of conditions define normal operation. First, the process must obey certain constraints $g(z) = 0$, shown as a surface in Fig. 3, which represent the governing equations of the process, such as mass balances or process dynamics. Second, the process can be characterized in terms of a range of normal variability within the subspace defined by the model equations, shown as a contour in Fig. 3.

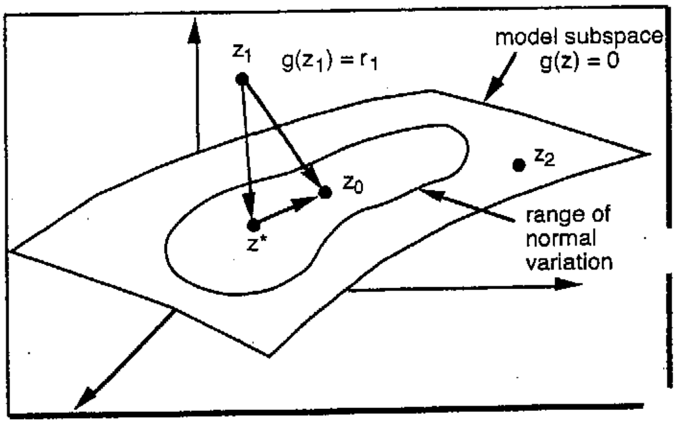

Figure 3. Fault detection.

Fault detection requires both the detection of departures from the model subspace, indicative of changes in the equations governing the process (e.g. the point z_1), and detection of excessive variability within the model subspace (e.g. point z_2), indicating an excursion from normal operating conditions. Different fault detection methods can be interpreted in terms of this general framework:

- *Alarm limits*. Alarms set at the safe operating bounds of measured variables respond only to the very large excursions from normal operation, and represent a particularly insensitive method of fault detection.
- *Univariate statistical tests*. Limit checks on individual variables, such as 3σ limits, collectively represent a hyperrectangle enclosing the normal region which may grossly overestimate the extent of the normal region, causing low sensitivity to faults.
- *Multivariate statistical tests*. Multivariate statistical models can form relatively accurate representations of the normal region. The Hotellings T^2 test, for example, defines a hyperellipse whose axes are small in dimensions orthogonal to the model subspace, while approximating variations within the model subspace.
- *Model residual tests*. Techniques based on evaluating model residuals $r = g(z)$, such as gross error detection methods and tests based on the innovations of (extended) Kalman filters, are capable of detecting departures from the model subspace. These tests, however, neglect variation within the model subspace.

- *Compound tests*. PCA and PLS, discussed below, are examples of methods that support compound tests involving both model violation and variability, though restricted to linear constraints on the features determined by regression of historical data. Similar techniques for general nonlinear analytical models are currently lacking.

The decision metrics associated with fault detection methods are frequently based on two auxiliary quantities: the rectified value z^* (see Fig. 3), defined as the most likely point in the model subspace given z (which, under typical assumptions, is the mapping of z to the closest point in the model subspace), and the mean value in the model subspace z_0. These metrics may include:

a. The distance from z to z_0, or the magnitude of the components $z_i - z_{i0}$;
b. The distance from z to z^*, or the magnitude of the components $z_i - z^*_i$;
c. The distance from z^* to z_0, often measured in a coordinate system defined in the model subspace.
d. The magnitude of the model residual r, or of its components, r_i. Because $r = A\ (z - z^*)$, where A is the local linearization of g(z) around z^*, this test is closely related to (b).

For example, the global and nodal tests involve metric (d), while a third gross error detection test, the measurement test, involves metric (b). PCA and PLS-based fault detection use both (b) and (c), while multivariate SPC tests use metric (a). For all fault detection tests, determining appropriate detection thresholds requires balancing sensitivity against false alarm rate.

Fault Isolation

As a rule, it is desirable to apply fault isolation to the maximum possible extent before attempting fault identification, due to the fact that *fault isolation can be performed without fault models, by performing fault detection over different plant subsystems*. As a result, fault isolation presents much more modest knowledge requirements than fault identification. By isolating the fault, it may be possible to formulate a response without further fault identification.

The three approaches to fault isolation are: disaggregation, hierarchical decomposition, and topological search. In the disaggregated approach, the process is decomposed into several non-interacting subsystems. Each subsystem requires a separate fault detection test, and the fault is located by continuously monitoring each subsystem for faults. This approach is suitable for processes where there are decoupled processing centers.

In the hierarchical approach, the plant is broken into successively smaller subsystems. Each level of aggregation requires a fault detection test, although not all subsystems need to be monitored continuously. When a fault is detected at the top level of the hierarchy, diagnostic focus can be shifted to inferior subsystems to refine the fault origin by applying fault detection to those subsystems. Examples of this approach include McDowell and Davis (1991), Miller, et al. (1994) and Ramesh, et al. (1992). It should be noted that this approach lends itself well to an object-oriented representation and an interface featuring "zooming in" on the fault origin through successive layers of detail.

The topological approach involves modeling subsystems and their causal precedence. Instead of assuming non-interacting subsystems, if faults appear in several subsystems, the subsystem farthest upstream is assumed to be the origin of the fault, since it is possible that the effects of the fault can propagate to subsequent subsystems. This approach can work even in the case of strongly connected subsystems (involving recycles or feedback loops), so long as there is sufficient time delay in the loop to make the fault origin clear before the effects of feedback obscure the origin. However, if the fault detection sensitivity in different subsystems are not equal, the point of earliest detection may not be the subsystem containing the fault.

Fault Identification

Fault identification requires information on how possible faults relate to observable symptoms (features). In diagnostic systems, we are interested in deducing faults from symptoms, which is opposite to the natural causal directionality of predictive models. Diagnosis is therefore an inverse problem, and as such, there are issues of solution uniqueness, solution conditioning (particularly with respect to model uncertainty), and the algorithmic efficiency of the inversion process. Modeling effort and accuracy is also of particular concern, since there are potentially a large number of fault modes. The literature contains a wealth of qualitative, logical and semi-quantitative model representations developed to help address these issues.

To better understand the motivations for non-numerical models in diagnosis, it is instructive to consider some potential limitations of direct parameter estimation as a means of "inverting" a numerical model to determine faults given observed variable trajectories. First, the parameter estimation problem may have more than one local optimum, particularly when the set of possible faults is large, includes sensor failures, or contains faults with similar effects. Parameter estimation will converge arbitrarily to one of the local optima, missing other solutions that may be more likely, or of interest as alternative hypotheses. Many qualitative models used in diagnosis are designed specifically to yield a ranked list of possible faults without the expense of global optimization. Second, modeling errors may have unpredictable effects on estimated parameter values. On the other hand, abstract or approximate models can be "right" in the sense of matching the actual fault behavior more often, by making less precise or even incomplete predictions, thus increasing robustness. Third, numerical estimation may be computationally intractable for on-line application. Fourth, the modeling effort involved with numerical modeling of all fault modes might be prohibitive. These factors account for the interest in causal, semi-quantitative, and similar model representations in fault diagnosis.

In general, there are three approaches to utilizing a predictive model for on-line diagnosis:

1. Invert the model using an algorithm specific to the model form, such as parameter estimation in the case of numerical models, graphical search in the case of causal networks, or belief updating in the case of Bayesian network.
2. Apply a general technique such as hypothesis-test or comparative simulation to effectively invert the model. In hypothesis-test, a conjecture is made about the identity of the fault, and the model is used to yield a prediction of the expected features, which is then compared to the observed symptoms. The process is repeated until the best match is found. Comparative simulation is similar, except that multiple models are run in parallel with the process.
3. Off-line, create a database of predicted symptoms for each possible fault, and use inductive learning techniques on this database to yield a pattern classifier, such as a neural network or decision tree. This approach is often referred to as "model compilation."

Of the three general approaches, the second is the least efficient. The appeal of the third approach is superior on-line efficiency. However, compiled forms of knowledge such as pattern classifiers provide a weak basis for explanation and are not useful for supporting ancillary tasks such as prognosis and correction. Therefore, in practice is may be desirable to use pattern classifiers as a supplement to, rather than a replacement for, predictive models.

Specific FDI Approaches

In this section, we review recent progress in five classes of approaches to FDI: statistical process control, parameter estimation, analytical redundancy, causal model analysis, and pattern recognition. These methods are summarized in Table 1 in terms of their approaches to feature generation, fault detection, and fault identification.

Statistical Process Control Approaches

Statistical process control techniques such as Shewhart charts, Cusum charts and Hotelling's T^2 are natural candidates for detecting faults. These methodologies utilize normal data to build a statistical characterization of the normal operating region of the process that can be used to detect abnormal events.

Projection techniques such as principal component analysis (PCA) and partial least squares (PLS) have recently attracted attention. In these approaches, fault detection is accomplished by establishing control limits on: (1) variation in projection (score) space, and (2) mapping distance between the measured point and the point projected onto the plane of constraints. Extensions to batch process have been investigated by Nomikos and MacGregor (1995) and Kourti, et al. (1995). Applications are discussed in Piovoso and Kosanovich (1993).

Table 1. Categories of FDI Approaches and Their Approaches to Feature Extraction, Fault Detection and Identification.

Approach	Features generated	Detection method	Identification method
SPC (including PCA, PLS)	*sample mean, range, scores, contributions*	*3σ limits, T^2, etc.*	*pattern recognition on scores, contributions*
Estimation (including parallel models)	*estimated parameters and states*	*parameter bounds, innovation statistics*	*likelihood ratio (parallel models), classification of parameters*
Analytical redundancy (incl. parity space, input observers)	*model equation residuals*	*statistical checks of residuals*	*pattern recognition or causal analysis of residuals*
Causal analysis (SDG, fault tree, belief net, etc.)	*discretized or fuzzified measure-ments and residuals*	*simple range checks*	*graphical, logical, and abductive approaches*
Pattern recognition (neural net, decision tree, rules, etc.)	*any features*	*membership in normal class (if represented)*	*classifier output*

Fault isolation using PCA and PLS can be carried out by decomposing global PLS models into separate blocks representing different process units (MacGregor, et al., 1994). These authors also introduce contribution plots as a way to analyze the measurement sources of abnormal behavior within each block. Although contribution plots do not strictly qualify as fault identification techniques since they do not identify the fault, they may help focus operator attention.

To carry out true diagnosis with pre-defined fault modes, fault data is used to build PCA or PLS models characterizing fault behavior, either pooling data from all faults or by building a separate PLS/PCA model for each fault. Fault identification is carried out via pattern recognition on the scores (for combined models), or by comparative model analysis (for separate fault models). Vinson, et al. (1994) have provided a very interesting critique of the combined approach using the Tennessee Eastman problem; their main difficulty was unique classification of the patterns in the score space.

The appeal of statistical process control approaches lies in their simplicity, rather than in any unique modeling or statistical properties they might possess. PCA and PLS are in essence regression techniques producing linear algebraic models that characterize normal operation. Thus PCA and PLS can be considered a simple case of analytical redundancy with linear algebraic models. Fault detection with PCA and PLS is also closely related to gross error detection techniques for linear systems, which have been extensively studied. Discussion of analogies between gross error detection and fault detection in PCA is given in Kramer (1992) and Kramer and Mah (1993).

Parameter and State Estimation Approaches

Faults associated with continuous parametric changes can be effectively diagnosed using parameter estimation techniques if the system is observable and appropriate mathematical models can be formulated. Beginning in the 1970's, many authors have applied extended Kalman filters and related approaches to this problem, recent examples including Li and Olson (1991), Fathi, et al. (1993), Isermann and Freyermuth (1991), Isermann (1993), and Ku, et al. (1992). To accomplish fault identification, many of these approaches apply pattern classification or causal analysis to the estimated states and parameters, which can be considered the features extracted in this approach.

Aside from the rather stringent modeling requirements, there are two main limitations in the parameter estimation approach. First, as the number of faults represented by undetermined parameters in the model grows large, observability may be violated. Second, structural (integer) parameters cannot typically be included in the estimation models. To overcome both limitations, it is necessary to use a bank of parallel estimators to reduce the number of adjustable parameters per model and/or replace structural parameterizations with explicitly enumerated structural alternatives. Not only does this multiply the modeling and computation work, but this strategy also introduces an additional model discrimination step (most often approached using the generalized likelihood ratio criterion) to determine which model best matches the process.

Approaches Based on Analytical Redundancy

Another large class of model-based methods is based on analytical redundancy. The basic approach is to compare actual behavior with that predicted by a model, and use the resulting differences (residuals) as feature inputs to fault identification via logical, causal, or pattern recognition techniques.

In the simplest form, measurements are directly substituted into model equations, and the resulting residual pattern is analyzed. The residual patterns are typically discretized or given a quantitative degree of violation before being causally related to faults. Recent examples of this type of approach include Chang, et al. (1994), Howell (1994), Lee (1994), Ning and Chou (1992), Petti and Dhurjati (1991), Petti, et al. (1990), and Tsai and Chou (1993). These methods differ mainly in terms of how process faults are associated with residual patterns, rather than the method of residual generation.

Some recent researchers, including Gertler and Singer (1990), Frank (1990), Frank and Ding (1994) and Patton and Chen (1993), have developed much more powerful methods, known as *structured residuals* and *unknown input observers*, that generate residuals with important properties such as optimal robustness to model uncertainty, decoupling from input disturbances, and incidence structures tailored to reveal the identity of specific faults. The theoretical basis of these methods, derived from traditional control and identification, is very sound. This line of work is advancing rapidly to include extensions to nonlinear processes, multiple faults, and structured model errors.

Approaches Based on Causal Analysis

Many contributions have been based on the concept of causal modeling of fault-symptom relationships. The relationships in these causal models have taken many different forms, including qualitative and semi-quantitative relationships, logical and probabilistic relations. Causal models have primarily been used for fault identification.

Graphical cause-and-effect models, exemplified by the signed directed graph (SDG), continue to appear frequently in the literature. In a SDG, nodes represent the system state variables and malfunctions, and arcs represent causal relationships. Recent work involves representing gains and delays, the use of fuzzy logic, diagnosis of multiple faults, increasing robustness and efficiency, and learning of fuzzy membership functions (see Chang and Yu, 1990; Finch, et al., 1990; Han, et al., 1994; Hsu and Yu, 1992; Mohindra and Clark, 1993; Wilcox and Himmelblau, 1994; Park and Seong, 1994; Qian, 1990; Yu and Lee, 1991).

Nuclear engineering has primarily adopted fault trees and similar representations for modeling causal knowledge. Because the relationship between faults and symptoms generally forms a graph, not a tree, the fault trees developed for each potential deviation or process alarm are not independent, and must be integrated through a pre-processing or run-time algorithm. Recent developments involve real-time use of fault trees (Gmytrasiewicz, et al., 1990; Zhang, 1994, Zhang, et al., 1994), goal trees and success trees (Chen and Modarres, 1992; Kim, et al., 1990; Nordvik, et al., 1994), and cause-consequence information generated from HAZOP or similar design-stage studies (Heino, et al., 1994; Martinez, et al., 1992).

Although many industrial diagnostic systems have incorporated elements of causal reasoning, from a theoretical viewpoint this area suffers from a multiplicity of modeling techniques and adoption of *ad hoc* criteria for identifying possible fault origins. Another persistent problem is the treatment of temporally-varying measurements. Because causal modeling deals with relating states of symbolic variables, the logical approach to unifying this area is through probability theory. Probability theory is sufficiently powerful to represent many types of causal influences, and also supports graphical analysis in the form of Bayesian belief networks (Rojas-Guzman and Kramer, 1993, 1994; Chu, 1993).

Pattern Recognition Methods

Pattern recognition uses associations between data patterns and fault classes without explicit modeling of internal process states or structure. Although model-based techniques are more flexible, there are several potential reasons for adopting a pattern recognition approach to fault identification:

- To capture human fault recognition rules and diagnostic associations that are not readily translated into mathematical or causal models;
- To capture the diagnostic information contained in fault data;
- As compilations of model-based descriptions for faster on-line response.

The first case suggests the application of rule-based systems. Although research interest in this approach has declined, diagnostic rules can nonetheless provide a compact and effective representation of simple diagnostic heuristics. The free-form character of this approach is both a potential benefit and a liability. Whether rule-based systems can progress beyond their current limited niche is an open question.

In contrast, the last five years has seen considerable research on training of pattern recognition systems from examples of fault behavior, addressing the second motivation given above. Since there is rarely any prior knowledge about the form of the probability distribution of the symptoms conditioned on the faults (e.g. Gaussian), non-parametric classifiers such as linear discriminants, nearest-neighbor methods, decision trees, or neural networks are usually applied. Although results obtained from these approaches are often similar in terms of accuracy, there is an advantage to methods that output the classification as a probability to permit inclusion of prior fault probabilities, control of false alarm rates, etc. Another important property to be maintained is the ability to detect novel situations for which the classifier is not trained.

Recent work in the application of neural networks to fault isolation include Becraft and Lee (1993), Fan, et al. (1993), Farell and Roat (1994), Hoskins, et al. (1991), Kavuri and Venkatasubramanian (1993, 1994), Kramer and Leonard (1990), Leonard and Kramer (1991), Marseguerra and Zio (1994), Sorsa and Kiovo (1991, 1993), Srinivasan and Batur (1994), Venkatasubramanian, et al. (1990), Watanabe, et al. (1994), and Xing and Okrent (1994). In spite of the high level of activity, the applicability of these methods is *seriously limited in practice by the availability of ample, representative, well-documented fault data.*

Decision trees are an alternative way to induce a classifier from training cases. Significant theory has been developed on the statistical interpretation and optimization of inductive decision trees in the presence of noisy features (Quinlan, 1990). Although originally derived for discrete feature vectors, decision trees can also be applied to a mixture of continuous and discrete inputs (Saraiva and Stephanopoulos, 1992). Decision trees possess certain advantages relative to neural networks, including the automatic selection of inputs and the transparency of the resulting classifier structure. However, the decision regions are limited to hyper-rectangular shapes.

For dynamic systems, pattern recognition in time is an important issue. The methods discussed so far identify faults given features developed from a moving time window of fixed length. Cheung and Stephanopoulos (1990), Konstantinov and Yoshida (1992), and Whiteley and Davis (1992) give methods for deriving qualitative features from dynamic trajectory. Wavelet transformations have also been suggested (Bakshi and Stephanopoulos, 1992). To introduce memory into the classification, an architecture like recurrent neural networks can be used, where the current classification is an input for future classifications. Alternately, Leonard and Kramer (1993) and Smyth (1994) present techniques which combine over time the instantaneous estimates of the classifier using knowledge of the statistical properties of the failure modes of the system.

Finally, pattern recognition can also be used to enhance run-time efficiency of model-based diagnosis by "compiling" the model (learning with data simulated using the model) since most classifiers run extremely fast once they have been trained. However, the compiled form will be much less maintainable than the model, and will not support functions like prognosis and explanation. An example of a system that compiles diagnostic knowledge is Far and Nakamichi (1993).

Architectures, Environments, and Applications

Thus far, we have discussed the theory and specific methods for fault diagnosis. In this section, we look at the integration of these building blocks into functional diagnostic systems, through a discussion of architectures, tools and environments, and applications.

Diagnostic System Architectures

Three R&D projects that have been carried out under the auspices of the European Commission exemplify the architectures of diagnostic systems: ARTIST, REAKT, and CommonKADS. Although by no means spanning the range of possibilities, these systems are representative of the state of the art, in which AI-derived methods are featured prominently.

The purpose of the ARTIST project (Leitch, et al., 1992) was to build a generic architecture for model-based diagnosis, and to verify the architecture on different applications. The architecture separates different types of knowledge: process knowledge, hypothesis generation knowledge, and diagnostic strategy knowledge. The resulting ARTIST architecture is shown in Fig. 4, where the role of the different modules are:

- *Predictor*: Produces behavioral predictions based on explicit models of the physical system from observations and detects discrepancies between observed and predicted behavior and/or different predictions.
- *Candidate proposer*: Generates diagnostic candidates based on discrepancies, ranks candidates according to some criterion, refines candidates with respect to structure and behavior, and discriminates between candidates.
- *Diagnostic strategist*: Controls the diagnostic process by evaluating the performance of the diagnostic process with respect to goals and resources, determining the foci of attention and suspicion, and determining the next diagnostic action.

ARTIST has been used in different applications, including diagnostic systems for the steam condenser and the boiler of thermal power plants (Angeli, et al., 1994).

In the REAKT project (Fjellheim, et al., 1994), an advanced tool for real time AI applications was developed. Its main features include a blackboard architecture, multiple cooperating agents, and predictable execution times for critical tasks. For the purpose of this paper, the main interest of REAKT lies in its support for a diagnostic/alarm handling application at an oil refinery. The general philosophy behind the application (called MORSAF) is to manage alarm situations as far as possible by *anticipation*, i.e. to compare actual with expected behavior. The expectations are alarms predicted by occurrence of previous alarms or "pre-alarm" situations. Of course, not all alarms can be anticipated, and unexpected alarms must be handled as well. A second principle of MORSAF is to base diagnosis of a fault (alarm) situation on *causal knowledge*, expressed in terms of causal networks. These networks are also used for prediction.

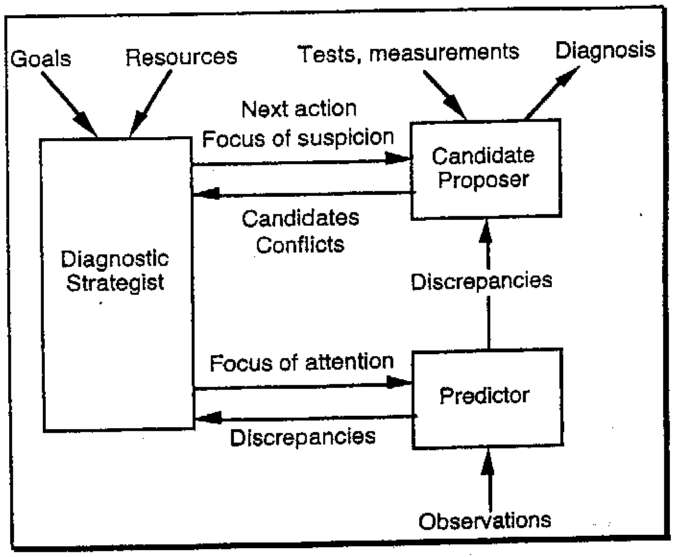

Figure 4. ARTIST architecture.

The overall architecture is illustrated by the flow diagram in Fig. 5. Incoming process data is analyzed by MORSAF, and used to detect potentially alarming situations, as well as updating the causal networks that represent state information to be used by causal diagnosis. Alarms are processed and filtered by matching earlier predicted alarms. Genuinely new (not expected) alarms are used to predict later alarms, while the expected ones (confirmed alarms) drive action suggestion. The latter also requires explanations provided by diagnosis, which is triggered by new alarms. The advice so generated is presented to the operator.

We include a brief description of CommonKADS (Schreiber, et al., 1994) here, because of its dominant position as a *de facto* standard methodology for development of knowledge based systems in Europe, and because the diagnostic task has been described in a systematic manner in this approach. A major theme in CommonKADS is *knowledge engineering as modeling*. In contrast to the more traditional view, where knowledge acquisition was seen as somehow "extracting" the knowledge from the head of an expert, CommonKADS stresses the active cooperation between the expert and the knowledge engineer in *modeling* the domain of expertise. The methods, notations and tools for supporting modeling are important ingredients of CommonKADS. Among several models, the *expertise model* is prominent. It contains domain knowledge, inference knowledge, and control knowledge.

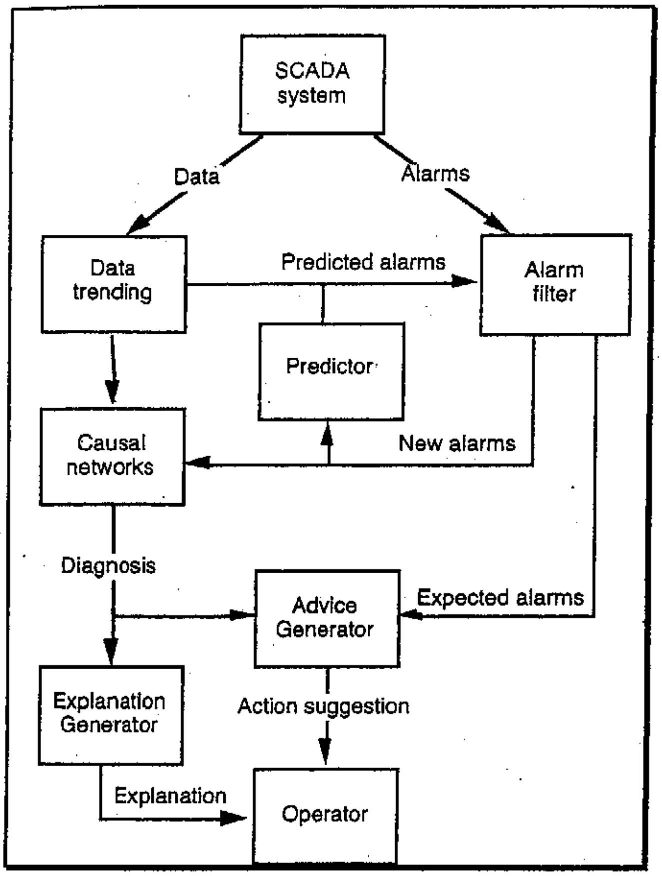

Figure 5. MORSAF architecture.

An extremely valuable part of CommonKADS is a library of generic and reusable Expertise Models (Breuker and Van de Velde, 1994). The library contains models for different types of diagnosis, including model-based diagnosis and "shallow" diagnosis (empirical, associational). With the help of the library, a new application need not start from scratch, but may rely on the structure provided by an appropriately selected library model for diagnosis. REAKT, mentioned above, embodies this philosophy in the domain of real-time diagnosis applications.

Tools and Environments

Implementation of diagnostic systems of the type outlined here requires high-level programming environments and powerful knowledge representation tools. The foundation needed to support these systems includes objects, methods, if-then rules, procedures, inheritance, hierarchies, relations, event triggering, and multi-tasking (for dealing with any number of problems simultaneously). Specific solution technologies, such as simulators, classifiers, identification packages, belief networks, statistical tools, fuzzy logic, etc. constitute a useful tool layer above the basic representations. Advanced intelligent automation systems environments, such as Gensym's G2, include representational and reasoning primitives (rules and objects), a procedural language, a graphical interface, a multi-threaded evaluation engine, and links to external code, databases, simulators, distributed control systems, and data historians.

Graphics also play a key role in how knowledge can be represented and used. Visual presentation can be enormously valuable; once gathered, reasoning about relationships in otherwise scattered data can have tremendous economic potential. In addition, many high-level solutions are most clearly represented by graphical languages. Connecting and configuring graphical objects expresses the desired behavior, and with the right software support, the graphic itself becomes the program. These are often combined with schematic representations of th system to be monitored. Gensym's G2 Diagnostic Assistant (Fraleigh, et al., 1992) and MFM (Larsson, 1994) typify graphical languages designed for diagnostic purposes.

Applications

Literally hundreds of industrial applications could be mentioned here, but only a few can be singled out for lack of space. Some excellent examples found in the literature include:

- Diagnosis of PreussenElectra's Staudinger power plant (Neupert and Schlee, 1994);
- Monitoring system for a cogeneration plant (Padalkar, et al., 1991);
- Operator support system for hydrogen peroxide production plant (Turunen, et al., 1992);
- Discharge reduction at a fertilizer plant (Saelid, et al., 1992).

The latter system exemplifies the potential benefits from diagnostic systems. Operating on one of Norsk Hydro's fertilizer plants since the beginning of 1993, the performance has been tracked and analyzed with very encouraging results. The introduction of the system led to a decrease of nitrogen discharge to the sewer from the ammonia stripper by a factor of ten, from an average of 9.6 kg/h to less than 1 kg/h.

Conclusions

As this paper has shown, the theoretical basis for computerized fault diagnosis has advanced considerably over the past few years. Still, this technology is not yet an established engineering discipline with a common terminology and a framework for systematically relating one's own problems with those reported by others. As a consequence, a practicing engineer will have no firm ground if given the assignment to design and implement a diagnosis system for a specific plant. He will be faced with an ill-defined problem, an array of available methods, and no reliable procedure for going from the problem to the solution. What is missing is a sound *engineering theory for diagnostic systems.*

On a more specific level, a number of open challenges remain within various approaches to diagnosis:

- Fault isolation is fundamentally important because it yields important diagnostic information without requiring models of faulty behavior. Topological, functional and compositional decomposition strategies for fault isolation need to be better characterized and their limits better understood.
- We would like to see better theoretical grounding of causal techniques through the application of probability theory, and integration of these methods with those derived from parameter estimation and analytical redundancy, thus profiting from the strong features of each.
- Techniques for determining the class of fault (as opposed to specific fault identity) have not received adequate study. Together with fault isolation techniques, fault classification could provide an attractive alternative to detailed identification based on pre-enumerated faults.
- Knowledge-based techniques are under-utilized due to the lack of understanding on how to integrate heuristic rules into maintainable, verifiable FDI systems.
- Because design-stage safety analyses such as HAZOP, cause-consequence analysis, etc. overlap many of the issues faced by monitoring and diagnostic systems, it seems reasonable to expect some re-use of information. Specifying a common knowledge format adequate for both off-line and on-line tasks should be a high priority.
- More attention should be paid to the ancillary problems of fault prediction, prognostication, and response. The theory in these areas lags considerably behind practical needs.

In terms of implementation, the requirements for success of diagnostic technology in industrial plants are not very different from those that apply to other operational systems. Some of the important issues are:

- The diagnostic system should not be separated from other tools used by the operator, by running on a separate computer, using a different user interface, etc. Instead, the system should be an integral part of the operator's normal working environment.
- The user interface must be up to the standards that one now takes for granted, i.e. intuitive graphics, zooming in/out on details.
- Models, in one way or another, will be part of diagnostic systems. High-level tools exist for modular construction, modification and reuse must be developed.
- Diagnostic systems will be critical to plant operations. User organizations will demand that they are delivered by vendors with a good track record, that are responsive to user needs, and provide support and maintenance services.

As we have shown earlier in this paper, a number of operational diagnosis systems have been successfully deployed and accepted by industrial users. However, many efforts have terminated with the shelving of unfinished prototypes. In general, computerized fault diagnosis systems are not as widely used today as one might expect, given the potential benefits. The reasons for this lack of acceptance, in our opinion, do not go much further than the issues listed in this section. If the research community is able to start developing engineering principles for building diagnostic systems, and vendors deliver these principles in robust, user-friendly software, user acceptance will follow. In an ever-more competitive business environment, industry has no choice other than adopting new technology that will give competitive advantage.

References

Angeli, F., L. Capetta, M. Gallant and L. Mazzocchi (1994). On-line performance and diagnosis of a steam condenser in a thermal power plant. *POWER-GEN Europe*. Cologne, Germany.

Bakshi, B.R. and G. Stephanopoulos (1992). Temporal representation of process trends for diagnosis and control. *IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*. Newark, DE. pp. 69-74.

Becraft, W.R. and P.L. Lee (1993). An integrated neural network/expert system approach for fault diagnosis. *Comput. Chem. Engng.*, **17**, 1001-1014.

Breuker, J. and W. Van de Velde (Eds.) (1994). *CommonKADS Library for Expertise Modeling*. IOS Press, Amsterdam.

Chang, C. and C. Yu (1990). On-line fault diagnosis using the signed directed graph. *Ind. Eng. Chem. Res.*, **29**, 1290-1299.

Chang, I-C., C. Yu and C. Liou (1994). Model-based approach for fault diagnosis. 1. principles of deep model algorithm. *Ind. Eng. Chem. Res.*, **33**, 1542-1555.

Chen, L.W. and M. Modarres (1992). Hierarchical decision process for fault administration. *Comput. Chem. Engng.*, **16**, 425-448.

Cheung, J.T.-Y. and G. Stephanopoulos (1990). Representation of process trends-I and II. *Comput. Chem. Engng.*, **14**, 495-539.

Chu, B. (1993). Fault diagnosis with continuous system models. *IEEE Trans. Sys. Man Cyber.*, **23**, 55-64.

Fan, J.Y., M. Niolaou and R.E. White (1993). An approach to fault diagnosis of chemical processes via neural networks. *AIChE J.*, **39**, 82-88.

Far, B.H. and M. Nakamichi (1993). Qualitative fault diagnosis in systems with nonintermittent concurrent faults: a subjective approach. *IEEE Trans. Sys. Man Cyber.*, **23**, 14-30.

Farell, A.E. and S.D. Roat (1994). Framework for enhancing fault diagnosis capabilities of artificial neural networks. *Comput. Chem. Engng.*, **18**, 613-635.

Fathi, Z., W.F. Ramirez and J. Korbicz (1993). Analytical and knowledge-based redundancy for fault diagnosis in process plants. *AIChE J.*, **39**, 42-56.

Finch, F.E., O.O. Oyeleye and M.A. Kramer (1990). A robust event-oriented methodology for diagnosis of dynamic process systems. *Comput. Chem. Engng.*, **14**, 1379-1396.

Fjellheim, R., T.B. Pettersen, B. Christoffersen and A. Nicholls (1994). Application methodology for REAKT systems. *2nd IFAC Symp. on Artif. Intell. in Real Time Control*. Valencia, Spain.

Fraleigh, S.P., F.E. Finch and G.M. Stanley (1992). Integrating dataflow and sequential control in a graphical diagnostic language. *IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*. Newark, DE. pp. 49-56.

Frank, P.M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy — a survey and some new results. *Automatica*, **26**, 459-474.

Frank, P.M. (1992). Robust model-based fault detection in dynamic systems. *IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*. Newark, DE. pp. 1-13.

Frank, P.M. and X. Ding (1994). Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, **30**, 789-804.

Gertler, J. and D. Singer (1990). A new structural framework for parity equation based failure detection and isolation. *Automatica*, **26**, 381-388.

Gmytrasiewicz, P., J.A. Hassberger and J.C. Lee (1990). Fault tree based diagnostics using fuzzy logic. *Trans. Patt. Anal. Mach. Intell.*, **12**, 1115-1119.

Han, C., R. Shih and L. Lee (1994). Quantifying signed directed graphs with the fuzzy set for fault diagnosis resolution improvement. *Ind. Eng. Chem. Res.*, **33**, 1943-1954.

Heino, P., I. Karvonen, T. Pettersen, R. Wennersten and T. Andersen (1994). Monitoring and analysis of hazards using HAZOP-based plant safety model. *Reliab. Eng. Sys. Safety*, **44**, 335-343.

Hoskins, J.C., K.M. Kaliyur and D.M. Himmelblau (1991). Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE J.*, **37**, 137-141.

Howell, J. (1994). Model-based fault detection in information poor plants. *Automatica*, **30**, 929-943.

Hsu, Y. and C. Yu (1992). A self-learning fault diagnosis system based on reinforcement learning. *Ind. Eng. Chem. Res.*, **31**, 1937-1946.

Isermann, R. and B. Freyermuth (1991). Process fault diagnosis based on process model knowledge — part 1: principles for fault diagnosis with parameter estimation. *Trans. ASME*, **113**, 620-626.

Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing — tutorial paper. *Automatica*, **29**, 815-835.

Kavuri, S.N. and V. Venkatasubramanian (1993). Representing bounded fault classes using neural networks with ellipsoidal activation functions. *Comput. Chem. Engng.*, **17**, 139-164.

Kavuri, S.N. and V. Venkatasubramanian (1994). Neural network decomposition strategies for large-scale fault diagnosis. *Int. J. Control*, **59**, 767-792.

Kim, I.S., M. Modarres and R.N.M. Hunt (1990). A model-based approach to on-line process disturbance management: the models. *Reliab. Eng. Sys. Safety*, **28**, 265-305.

Kim, I.S. (1994). Computerized systems for on-line management of failures: a state-of-the-art discussion of alarm systems and diagnostic systems applied in the nuclear industry. *Reliab. Eng. Sys. Safety*, **44**, 279-295.

Konstantinov, K.B. and T. Yoshida (1992). A method for on-line reasoning about the time-profiles of process variables. *IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*. Newark, DE. pp. 93-98.

Kourti, T., P. Nomikos and J.F. MacGregor (1995). Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *J. Process Control (to appear)*.

Kramer, M.A. (1992). Autoassociative neural networks. *Comput. Chem. Engng.*, **16**, 313-328.

Kramer, M.A. and J.L. Leonard (1990). Diagnosis using backpropagation neural networks: analysis and criticism. *Comput. Chem. Engng.*, **14**, 1323-1338.

Kramer, M.A. and R.S.H. Mah (1993). Model-based monitoring. In *Proc. Second Conf. on Foundations of Computer-Aided Process Operations*. CACHE Austin, TX. pp. 45-68.

Ku, W., R.H. Storer and C. Georgakis (1992). Disturbance detection and identification in statistical process control. *AIChE Natl. Meeting*. Miami Beach, FL.

Larsson, J.E. (1994). Diagnostic reasoning strategies for means-end models. *Automatica*, **30**, 775-787.

Lee, S.C. (1994). Sensor value validation based on systematic exploration of the sensor redundancy for fault diagnosis KBS. *IEEE Trans. Sys. Man Cyber.*, **24**, 594-605.

Leitch, R., H. Freitag, Q. Shen, P. Struss and G. Tornielli (1992). ARTIST: a methdological approach to specifying model based diagnostic systems. In Guida and Stefanini (Eds.), *Industrial Applications of Knowledge-Based Diagnosis*.

Leonard, J.A. and M.A. Kramer (1991) Radial basis function networks for classifying process faults. *IEEE Control Systems*, **11**, 31-38.

Leonard, J.A. and M.A. Kramer (1993). Diagnosing dynamic faults using modular neural nets. *IEEE Expert*, **8**(2) 44-53.

Li, R. and J.H. Olson (1991). Fault detection and diagnosis in a closed-loop nonlinear distillation process: application of extended Kalman filters. *Ind. Eng. Chem. Res.*, **30**, 898-908.

MacGregor, J.F., C. Jaeckle, C. Kiparissides and M. Koutoudi (1994). Process monitoring and diagnosis by multiblock PLS methods. *AIChE J.*, **40**, 826-838.

Marseguerra, M. and E. Zio (1994). Fault diagnosis via neural networks: the Boltzmann machine. *Nucl. Sci. Engng.*, **117**, 194-200.

Martinez, E., L. Beltramini, H. Leone, C.A. Ruiz and E. Huete (1992). Knowledge elicitation and structuring for a real-time expert system monitoring a butadiene extraction system. *Comput. Chem. Engng.*, **16**, S345-S352.

McDowell, J.K. and J.F. Davis (1991). Managing qualitative simulation in knowledge-based chemical diagnosis. *AIChE J.*, **37**, 569-580.

Miller, D.W., J.W. Hines, B.K. Hajek, L. Khartabill, C.R. Hardy, M.A. Haas and L. Robbins (1994). Experience with the hierarchical method for diagnosis of faults in nuclear power plant systems. *Reliab. Eng. Sys. Safety*, **44**, 297-311.

Mohindra, S. and P.A. Clark (1993). A distributed fault diagnosis method based on digraph models: steady-state analysis. *Comput. Chem. Engng.*, **17**, 193-210.

Neupert, D. and M. Schlee (1994). Staudinger power plant uses expert control system to increase efficiency. *Power Engng. Intl.*, Nov/Dec.

Ning, J.N. and H.P. Chou (1992). Construction and evaluation of fault detection network for signal validation. *IEEE Trans. Nucl. Sci.*, **39**, 943-947.

Nomikos, P. and J.F. MacGregor (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics*, **37**, 41-59.

Nordvik, J.P., N. Mitchison and M. Wilkens (1994). The role of the goal tree-success tree model in the real-time supervision of hazardous plants. *Reliab. Eng. Sys. Safety*, **44**, 345-360.

Padalkar, S., G. Karsai, C. Biegl, J. Sztipanovits, K. Okuda and N. Miyasaka (1991). Real-time fault diagnosis. *IEEE Expert*, **6**(3), 75-85.

Park, J.H. and P.H. Seong (1994). Nuclear power plant pressurizer fault diagnosis using fuzzy signed-digraph and spurious faults eliminations methods. *Ann. Nucl. Energy*, **21**, 357-369.

Patton, R.J. and J. Chen (1993). Optimal unknown input disturbance matrix selection in robust fault diagnosis. *Automatica*, **29**, 837-841.

Petti, T.F. and P.S. Dhurjati (1991). Object-based automated fault diagnosis. *Chem. Eng. Comm.*, **102**, 107-126.

Petti, T.F., J. Klein and P.S. Dhurjati (1990). Diagnostic model processor: using deep knowledge for process fault diagnosis. *AIChE J.*, **36**, 565-575.

Piovoso, M.J. and K.A. Kosanovich (1993). Applications of multivariate statistical methods to process monitoring and controller design. *Int. J. Control*, **59**, 743-765.

Qian, D.Q. (1990). An improved method for fault location of chemical plants. *Comput. Chem. Engng.*, **14**, 41-48.

Quinlan, J.R. (1990). Decision trees and decisionmaking. *IEEE Trans. on Sys. Man Cybernetics*, **20**, 339-346.

Ramesh, T.S., J.F. Davis and G.M. Schwenzer (1992). Knowledge-based diagnostic systems for continuous process operations based upon the task framework. *Comput. Chem. Engng.*, **16**, 109-127.

Rojas-Guzmán, C. and M.A. Kramer (1993). Comparison of belief networks and rule-based expert systems for fault diagnosis of chemical processes. *Engng. Applic. Artif. Intell.*, **6**, 191-202.

Rojas-Guzmán, C. and M.A. Kramer (1994). Multi-stage Bayesian networks subsume digraph and residual-pattern approaches to fault diagnosis. *Proc. 5th Intl. Symp. on Process Systems Engng.* Kyongju, Korea. pp. 947-952.

Saelid, S., A. Mjaavatten and K. Fjalestad (1992). An object oriented operator support system based on process models and an expert system shell. *Comput. Chem. Engng.*, **16**, S97-S108.

Saraiva, P. and G. Stephanopoulos (1992). Continuous process improvement through inductive and analogical learning. *AIChE J.*, **38**, 161-183.

Schreiber, G., B. Wielinga, R. de Hoog, H. Akkermans and W. Van de Velde (1994). CommonKADS: a comprehensive methodology for KBS development. *IEEE Expert*, **9**(6), 28-37.

Smyth, P. (1994). Hidden Markov models for fault detection in dynamic systems. *Pattern Recog.*, **27**, 149-164.

Sorsa, T. and H.N. Koivo (1991). Neural networks in process fault diagnosis. *IEEE Trans. Sys. Man Cyber.*, **21**, 815-825.

Sorsa, T. and H.N. Koivo (1993). Application of artificial neural networks in process fault diagnosis. *Automatica*, **29**, 843-849.

Srinivasan, A. and C. Batur (1994). Hopfield/ART-1 neural network-based fault detection and isolation. *IEEE Trans. Neural Networks*, **5**, 890-899.

Stephanopoulos, G. and C. Han (1994). Intelligent systems in process engineering: a review. *Proc. 5th Intl. Symp. on Process Systems Engineering*. Kyongju, Korea. pp. 1339-1366.

Tsai, T.M. and H.P. Chou (1993). Sensor fault detection with single sensor parity relation. *Nucl. Sci. Engng.*, **114**, 141-148.

Turunen, I., M. Piironen and K. Westerstrahle (1992). Expero — an advanced support system for hydrogen peroxide process control. *Comput. Chem. Engng.*, **16**, S531-S538.

Venkatasubramanian, V., R. Vaidyanathan and Y. Yamamoto (1990). Process fault detection and diagnosis using neural networks-I. steady-state processes. *Comput. Chem. Engng.*, **14**, 699-712.

Vinson, J.M., L.H. Ungar and R.D. DeVeaux (1994). Using PLS for fault analysis. *AIChE Natl. Meeting*. San Francisco, CA.

Watanabe, K., S. Hirota, L. Hou and D.M. Himmelblau (1994). Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks. *AIChE J.*, **40**, 839-848.

Whiteley, J.R. and J.F. Davis (1992). Qualitative interpretation of sensor patterns using a similarity-based method. *IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*. Newark, DE. pp. 75-80.

Wilcox, N.A. and D.M. Himmelblau (1994). The possible cause and effect graphs model for fault diagnosis. *Comput. Chem. Engng.*, **18**, 103-128.

Xing, L. and D. Okrent (1994). The use of neural network and a prototype expert system in BWR ATWS accidents diagnosis. *Reliab. Eng. Sys. Safety*, **44**, 361-372.

Yu, C.C. and C. Lee. (1991). Fault diagnosis based on qualitative/quantitative knowledge. *AIChE J.*, **37**, 617-628.

Zhang, Q. (1994). A frequency and knowledge tree/causality diagram based expert system approach for fault diagnosis. *Reliab. Eng. Sys. Safety*, **43**, 17-28.

Zhang, Q., X. An, J. Gu, B. Zhao, D. Xu and S. Xi (1994). Application of FBOLES — a prototype expert system for fault diagnosis in nuclear power plants. *Reliab. Eng. Sys. Safety*, **44**, 225-235.