Available online at www.sciencedirect.com

SCIENCE DIRECT

Computers and Chemical Engineering 27 (2003) 313–326

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

A review of process fault detection and diagnosis Part II: Qualitative models and search strategies

Venkat Venkatasubramanian [a,*], Raghunathan Rengaswamy [b,*], Surya N. Kavuri [c]

[a] *Laboratory for Intelligent Process Systems, School of Chemical Engineering, Purdue University, West Lafayette, IN 47907, USA*
[b] *Department of Chemical Engineering, Clarkson University, Potsdam, NY 13699-5705, USA*
[c] *BP, Houston, TX, USA*

Received 12 February 2001; accepted 22 April 2002

Abstract

In this part of the paper, we review qualitative model representations and search strategies used in fault diagnostic systems. Qualitative models are usually developed based on some fundamental understanding of the physics and chemistry of the process. Various forms of qualitative models such as causal models and abstraction hierarchies are discussed. The relative advantages and disadvantages of these representations are highlighted. In terms of search strategies, we broadly classify them as topographic and symptomatic search techniques. Topographic searches perform malfunction analysis using a template of normal operation, whereas, symptomatic searches look for symptoms to direct the search to the fault location. Various forms of topographic and symptomatic search strategies are discussed.

Keywords: Symptomatic search; Topographic search

1. Introduction

Diagnostic activity comprises of two important components: a priori domain knowledge and search strategy. The basic a priori knowledge that is needed for fault diagnosis is a set of failures and the relationship between the observations (symptoms) and the failures. A diagnostic system may have them explicitly (as in a table look-up), or it may be inferred from some source of domain knowledge. A priori domain knowledge may be developed from a fundamental understanding of the process using first-principles knowledge. Such knowledge is referred to as deep, causal or model-based knowledge (Milne, 1987). On the other hand, it may be gleaned from past experience with the process. This knowledge is referred to as shallow, compiled, evidential or process history-based knowledge.

The model-based a priori knowledge can be broadly classified as qualitative or quantitative. The model is usually developed based on some fundamental understanding of the physics of the process. In quantitative models this understanding is expressed in terms of mathematical functional relationships between the inputs and outputs of the system. In contrast, in qualitative models these relationships are expressed in terms of qualitative functions centered around different units in a process. The qualitative models can be developed either

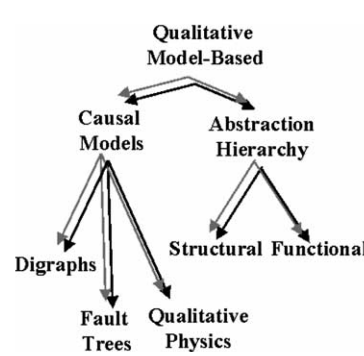

Fig. 1. Forms of qualitative knowledge.

* Corresponding authors. Tel.: +1-765-494-0734; fax: +1-765-494-0805 (V. Venkatasubramanian); Tel.: +1-315-268-4423; fax: +1-315-268-6654 (R. Rengaswamy).

E-mail addresses: venkat@ecn.purdue.edu (V. Venkatasubramanian), raghu@clarkson.edu (R. Rengaswamy).

0098-1354/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0098-1354(02)00161-8

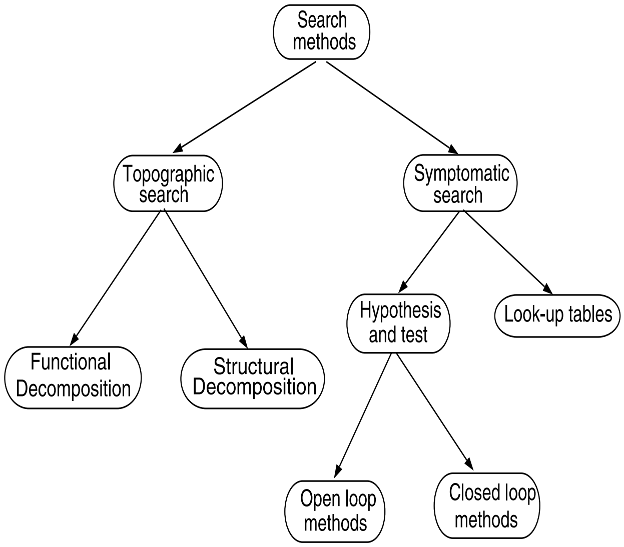

Fig. 2. Classification of search strategies.

as qualitative causal models or abstraction hierarchies. Fig. 1 shows the taxonomy of domain knowledge based on these two broad categories. In this part of the paper we will review the various qualitative knowledge forms shown in Fig. 1.

There are fundamentally two different approaches to search in fault diagnosis (Rasmussen, 1986): topographic search and symptomatic search. Topographic searches perform malfunction analysis using a template of normal operation, whereas, symptomatic searches look for symptoms to direct the search to the fault location. Fig. 2 shows a classification of diagnostic systems based on the search methods they employ. In this paper we discuss the various search methods as shown in Fig. 2.

2. Qualitative models

The development of knowledge-based expert systems was the first attempt to capture knowledge to draw conclusions in a formal methodology. An expert system is a computer program that mimics the cognitive behavior of a human expert solving problems in a particular domain. It consists of a knowledge base, essentially a large set of if–then–else rules and an inference engine which searches through the knowledge base to derive conclusions from given facts. Also, the tree of these if–then–else clauses grows rapidly with the behavioral complexity of the system. The problem with this kind of knowledge representation is that it does not have any understanding of the underlying physics of the system, and therefore fails in cases where a new condition is encountered that is not defined in the knowledge base. Therefore, this kind of knowledge is referred to as 'shallow' since it does not have a deep, fundamental understanding of the system.

In symbolic reasoning, one often addresses three different kinds of reasoning. They are *abductive*, *inductive* and *default* reasonings. Abduction is the generation of a hypothetical explanation (or cause) for what has been observed. Unlike simple logical deduction, we can get more than one answer in abductive reasoning. Since there is no general way to decide between alternatives, the best one can do is to find a hypothesis that is most probable. Thus abduction can be thought of as reasoning where we weigh the evidences in the presence of uncertainty. Searching for the cause of an abnormality in a process system is thus an abductive reasoning. In MODEX2 (Venkatasubramanian & Rich, 1988), a model based expert system for fault diagnosis, abductive reasoning is used to generate hypotheses for the sources of faults. In addition, abduction also provides explanations of how the cause could have resulted in the abnormality observed. Such a facility is useful in providing decision support to plant operators. Use of knowledge representation matters a great deal in determining the computational effort. Model based reasoning allows for efficient bottom–up abduction by suggesting proper rules to check. Efficiency of such bottom–up search in abduction is considerable (Charniak & McDermott, 1984).

Early work in learning concentrated on systems for pattern classification and game playing. Inductive learning is the classification of a set of experiences into categories or concepts. Inductive learning is performed when one generalizes or specializes a concept definition learned so that it includes all experiences that belong to that concept and exclude those that do not. A clear definition of a concept or category is rarely simple because of the great variety of experiences and uncertainty (noisy data or observations). For this reason, one prefers an adaptive learning scheme. An example of an adaptive learning scheme is the failure-driven learning. Failure-driven learning is refining a concept from failures of expectations as one has related experiences. The failure of heuristic judgment in detecting a source of malfunction in fault diagnosis can trigger a change in the knowledge (or rule) that resulted in the judgment (Rich & Venkatasubramanian, 1989). Experiences with abnormalities in a plant can be used to generate rules that relate a set of observations with specific causes. One can refine this experiential knowledge over time by generalizing to successful cases not covered and specializing when exceptions are noticed.

One frequently makes default assumptions on the values of various quantities that are manipulated, with the intention of allowing specific reasons for other values to override the current values (e.g. since the outlet is blocked, the flow is now zero), or of rejecting the default if it leads to an inconsistency (e.g. since the outlet of the tank is blocked, there cannot be a decrease in tank level). A fundamental feature of default reasoning is that it is non-monotonic. In traditional logic, once a fact is deduced, it is considered to remain true for the rest of the reasoning. This is what one means by monotonic. However, as new evidence arises, one often needs to revise the deduced facts to maintain logical consistency. Let us consider our previous argument where we deduced that the tank level cannot decrease (since the outlet of the tank is blocked). After this deduction, if we get new evidence that the tank has a large leak, we will have to retract the conclusion that the tank level cannot decrease. Such a reasoning where retraction of deductions is allowed is non-monotonic. Default reasoning or non-monotonic reasoning is an invaluable tool in dealing with situations where all the information is not available at a time or if one has to reason about many, probably inconsistent, cases simultaneously. Reiter (1987) has shown how default logic can be used for reasoning about multiple faults or causes for an abnormality.

The need for a reasoning tool which can qualitatively model a system, capture the causal structure of the system in a more profound manner than the conventional expert systems and yet be not as rigid in nature as numeric simulation led to development of many methodologies to qualitatively represent knowledge, and to reason from them. In this section we will discuss these various forms of qualitative knowledge.

2.1. Digraphs based causal models

Diagnosis is the inverse of simulation. Simulation is concerned with the derivation of the behavior of the process given its structural and functional aspects. Diagnosis, on the other hand, is concerned with deducing structure from the behavior. This kind of deduction needs reasoning about the cause and effect relationships in the process. In the evidential reasoning approach to diagnosis, heuristic information in the form of production rules is used. The underlying cause–effect relationships of the process are implicit in this form of reasoning. In the first-principles model-based approach, one begins with a description of the system together with the observations made from the malfunctioning process. The reasoning here is to identify functional changes which resulted in the malfunctioning of the process (Davis, 1984; Rich & Venkatasubramanian, 1987; Venkatasubramanian & Rich, 1988). It is in the later approach that qualitative causal models are very important and are used extensively.

Cause–effect relations or models can be represented in the form of signed digraphs (SDG). Digraph is a graph with directed arcs between the nodes and SDG is a graph in which the directed arcs have a positive or negative sign attached to them. The directed arcs lead from the 'cause' nodes to the 'effect' nodes. Each node in the SDG corresponds to the deviation from the steady state of a variable. SDGs have nodes which represent events or variables and edges which represent the relationship between the nodes. They are much more compact than truth tables, decision tables, or finite state models. To understand digraphs, consider a tank where F_1 is the inlet flow, F_2 is the outlet flow, and Z is the height of the liquid in a tank. The equations that represent this system are:

$$F_1 - F_2 = \frac{dZ}{dt}$$

$$F_2 = \frac{Z}{R}$$

A corresponding digraph is given in Fig. 3. The figure can be read as follows: an external change causes the flowrate F_1 to change, this causes a change in the liquid level in the tank (dZ and Z), this in turn causes the outlet flowrate F_2 to change and this in turn causes the liquid level to change (a feedback loop here). The signs in the arcs represent the direction of change. In a general situation, the arcs may be event dependent, i.e., the relationship between two events or variables may be dependent on other events or variables in the system.

SDGs provide a very efficient way of representing qualitative models graphically. There are mainly three kind of nodes in a typical SDG representing a chemical process: (a) those with only output arcs from them. They represent basic or more precisely fault variables which can change independently; (b) those which have both input and output arcs, most often called process variables and (c) those with input arcs only. They are often called output variables and they do not affect any other variable.

SDGs have been the most widely used form of causal knowledge for process fault diagnosis. Hence we will review the important contributions in the field of SDG representation in sufficient detail. At this juncture, a brief mention of terms used in digraph analysis is in order. A subset of a digraph is called a strongly connected component (SCC) if every node of this subset can be reached from every other node of this subset. Maximal strongly connected component (MSCC) in a digraph is a node or SCC with no input arcs to it. MSCC is sometime also called root node.

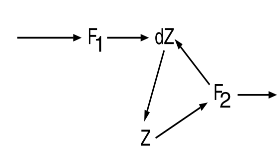

Fig. 3. Digraph for a simple tank example.

Iri, Aoki, O'Shima, and Matsuyama (1979) were the first to use SDG for fault diagnosis. SDG can be obtained either from the mathematical model of the underlying process or from the operational data (operator's experience). From SDG, they derive what is called a cause–effect graph (CE graph). CE graph consists of only valid nodes (nodes which are abnormal) and consistent arcs. Consistent arcs are the arcs which potentially explain local propagation of the fault and hence the observed symptom or pattern. Only valid nodes are considered because nodes which are normal do not provide any path from sensor nodes to the fault nodes. Sign of nodes in a SDG constitutes a pattern. When the sign of some of the nodes is not known then the pattern is called a partial pattern. In a typical process, all the process variables are not usually measured. When some of the nodes show abnormality, a CE graph with partial pattern (known as quasi-CE graph) is considered for diagnosis. The sign of the unmeasured nodes is assumed sequentially and the quasi-CE graph is expanded. All possible MSCCs are identified as potential fault nodes by propagation through the CE graph. When combinatorial search space for the sign of unmeasured nodes is exhausted, the diagnostic reasoning stops.

Umeda, Kuriyama, O'shima, and Matsuyama (1980) showed how SDG can be obtained from differential algebraic equations for the process. In this work the arc gains were allowed to vary dynamically—essentially to handle nonlinearities. Shiozaki, Matsuyama, O'Shima, and Iri (1985) address the issue of conditional arc in their SDG representation. In their work, conditional or dual event is represented by two arcs of different signs. In this manner, the original causal origin will never be omitted though the number of spurious solutions might increase. Shiozaki, Matsuyama, Tano, and O'Shima (1985) also extend the idea of SDG to include five-range patterns instead of the usual three-range pattern used in the standard SDG. This extension allows one to apply SDG even in the absence of detailed quantitative information which is crucial for setting the thresholds in a three-range pattern SDG.

Kokawa, Satoshi, and Shigai (1983) used partial system dynamics (such as time required for fault propagation from one variable/process unit to another), statistical information about equipment failure, e.g., failure probability, and digraphs to represent the failure propagation network for identifying fault location. No sign is required in this analysis but the method is limited to systems without any feedback.

Rule based method using SDG has been used for fault diagnosis by Kramer and Palowitch (1987). The basic idea here is to consider all possible simulation trees. These simulation trees consist of directed paths from root (fault) nodes to symptom nodes. The simulation tree is converted into logical rules. A methodology has been presented for direct derivation of logical rules from SDG. SDG used in these approaches is similar to the one developed using the approach suggested by Umeda et al. (1980).

An important work in the field of steady state qualitative simulation (QSIM) using SDG has been presented by Oyeleye and Kramer (1988). In this work, algorithms for the systematic use of steady state as well as dynamic model equations for digraph generation is discussed. The major focus of this work is on elimination of spurious solutions (generated due to qualitative nature of equations) without losing completeness. Other than the qualitative equations or confluences generated from the SDG, latent or redundant equations are obtained from the original steady state equations through algebraic manipulation. In addition to this, the SDG generated from the dynamic model is used to get some more confluences which turn out to be very helpful in eliminating spurious solutions.

Confluences from latent equations are generated from original set of independent equations through algebraic manipulation. The new equations are called latent (or sometimes redundant) equations as they do not provide any new information for quantitative analysis. These equations are very helpful in reducing the number of spurious solutions. Some heuristics (such as sub-global and global balances, elimination of group of variables etc.) could be used for generating potentially useful latent equations. There are some limitations imposed by difficulties in automation (or modularity) and amount of quantitative information required in generating latent equations. This type of confluences are sometimes called non-causal confluences because in general algebraic equations are unable to represent causality explicitly.

Confluences describing the dynamics in systems represented by ODEs or PDEs can be derived, as causality is explicit, and it is from right to left. Directed arcs are drawn from the variables on the RHS to the variables on LHS (Iri et al., 1979). The SDG thus obtained, can be used for analyzing the evolution of the system for faults. It has been shown that earlier methods fail to predict the ultimate response of the system whenever inverse response (IR) and/or compensatory response (CR) is exhibited by the system (except for control loops). This is because the earlier methods assume that fault can propagate through simple causal paths (SCPs) only. No node is repeated in a SCP. IR/CR occurs due to the presence of multiple feed forward paths from root node to the node under consideration with conflicting effects, or due to existence of negative feedback (control as well as non-control) loops. The corresponding variables are called inverse and compensatory variables (IVs and CVs) respectively. Controlled variables are just a special case of CVs. Similar to controlled variables, CVs can pass disturbances without deviation in themselves. Necessary conditions under which system can exhibit IR/CR can be derived. Algorithms for identifying IVs and CVs have been given (Oyeleye & Kramer, 1988). Sufficiency conditions for IVs can be derived from quantitative analysis only (Chang & Yu, 1990). To propagate the effect across IVs and CVs, additional arcs need to be drawn across IVs and CVs. Once all the additional arcs are drawn, the resulting SDG is called the Extended SDG (ESDG) (Oyeleye & Kramer, 1988).

Once ESDG is obtained, assumption of SCP can be used to predict the steady state response. ESDG can be converted into equivalent set of confluences using nodal balance (on nodes with at least one input arc). Nodal balance simply states that the sign of a node is the net influence of arcs incident on that node. These confluences are causal in nature. Some of these are the same as the non-causal confluences, but a few of them are new and it could be very difficult to derive these confluences as latent equations. Some of these confluences help in eliminating the unstable steady state solutions. In negative feedback loops, nodal balance on ESDG automatically ensures the presence of SCPs (only SCPs) because invoking a complex path results in violation of at least one nodal balance at steady state. Thus complex causal paths are automatically eliminated. It turns out that nodal balances alone are unable to eliminate complex paths in positive feedback loops and hence spurious solutions are generated. To avoid this, a restriction that at least one of the nodes in positive feedback loop should assume the sign depicted by an external effect is imposed. However, in this work, no formal metrics on the efficiency with which spurious solutions can be removed is discussed as this is a difficult issue and the practical utility of such an analysis might be questionable given the fact that the results derived might be too case-specific.

In somewhat similar work, Chang and Yu (1990) have reported various techniques that are useful in simplifying SDGs for fault diagnosis. Special attention is given to control loops. They have shown that ambiguities in the sign of arcs in control loops can be resolved to a great extent by writing the controller equations in discretized and velocity from. The resulting SDG is converted into equivalent rules as discussed by Kramer and Palowitch (1987) and solved. The methodology has been shown to be better than previous methods for fault diagnosis of a CSTR system. In recent years, Wilcox and Himmelblau (1994a,b) have approached the problem of fault diagnosis using what is known as possible cause and effect graph (PCEG) models. PCEG is in some sense a digraph. PCEG models inherit a number of properties from SDGs such as ease of construction, handling recycle systems, control loops and completeness. In addition, PCEG models provide more accurate information about the state of the system and thus a potential reduction in search space is achieved.

Vaidhyanathan and Venkatasubramanian (1995) have used digraph-based models for automated HAZOP analysis. HAZOP-digraph models (HDG) are used to accomplish the task. HDGs can be seen as extension of standard SDGs. In addition to process variable nodes and arcs corresponding to the SDG, HDG has abnormal cause and adverse consequence nodes containing the knowledge required for hazard identification useful for HAZOP analysis. These nodes can be attached to the process variables. HAZOP analysis is performed by generating deviations in process variables by using appropriate guide words such as LESS OF, MORE OF etc. and finding all possible adverse consequences by propagating through HDGs. Use of SDGs for multiple fault detection is demonstrated by Vedam and Venkatasubramanian (1997). The poor resolution of SDG is overcome by using a knowledge base consisting of knowledge about the process constraints, maintenance schedule and so on.

Improvement of fault resolution in SDG models through the use of fuzzy set theory is discussed by Han, Shih, and Lee (1994). In their approach, after the strong components which are the possible fault origins are located, quantitative fuzzy set manipulation is introduced and the variables are sequentially arranged by their degree of membership and the most probable fault origins are located. The results show that such an approach improves the accuracy of resolution. Use of fuzzy logic principles with SDGs for the removal of spurious solutions is discussed in Shih and Lee (1995a,b). In their work, dynamic confluences are converted to dynamic fuzzy relations. Using this fuzzy graph, spurious interpretations attributable to system compensations and IRs from feedback loops are eliminated. This method also applies fuzzy reasoning to estimate the state of unmeasured variables to explain fault propagation paths. A number of other researchers have looked at the role of fuzzy reasoning in qualitative models. Tarifa and Scenna (1997) have discussed the combined use of SDG and fuzzy reasoning for fault diagnosis. Work has been done on qualitative process modelling using fuzzy digraphs and fuzzy causal relationships (Kim & Lee, 1998; Wang, Yang, Veloso, Lu, & McGreavy, 1995). Genovesi, Harmand, and Steyer (1999) have presented a framework for process supervision using fuzzy logic-based fault diagnosis. Li and Wang (2001) have presented how fuzzy digraphs can be used for qualitative and quantitative simulation of temporal behavior of process systems. The literature on combining fuzzy logic and qualitative models looks at improving the representational scope of qualitative models by increasing the granularity through the use of fuzzy representations of real-valued functions. In that sense, these hybrid approaches seem to hold promise.

2.2. Fault trees

Fault trees are used in analyzing the system reliability and safety. Fault tree analysis was originally developed at Bell Telephone Laboratories in 1961. Fault tree is a logic tree that propagates primary events or faults to the top level event or a hazard. The tree usually has layers of nodes. At each node different logic operations like AND and OR are performed for propagation. Fault-trees have been used in a variety of risk assessment and reliability analysis studies (Ulerich and Powers, 1988). A general fault tree analysis consists of the following four steps: (i) system definition; (ii) fault tree construction; (iii) qualitative evaluation and (iv) quantitative evaluation (Fussell, 1974). Before the construction of the fault tree, the analyst should possess a complete understanding of the system. In fact, a system description is also a part of the analysis documentation (Bennetts, 1974).

The fault tree is constructed by asking questions such as what could cause a top level event. In answering this question, one generates other events connected by logic nodes. The tree is expanded in this manner till one encounters events (primary events) which need not be developed further (Lapp & Powers, 1977). Once the fault tree is constructed, the next step in the analysis is the evaluation of the fault tree. Qualitative evaluation is concerned with the development of minimal cut sets, defined as being 'a collection of primary failures all of which are necessary and sufficient to cause the system failure by the minimal cut-set in question' (Fussell, 1974). A minimal cut-set identifies the critical component failures. In quantitative evaluation, knowledge about the probability of occurrence of primary events is used to calculate the probability of failure of the top level event. The term evaluation as used here is the assessment of probability of occurrence of top level events and not the evaluation of the exactness of the fault tree generated.

Fault trees provide a computational means for combining logic to analyze system faults. The attractive nature of fault tree stems from the fact that different logic nodes can be used (OR, AND, XOR) instead of the predominantly OR node used in the digraphs. This helps in reducing the spurious solutions and representing the system in a concise manner. The biggest problem with fault trees though is that the development is prone to mistakes at different stages. The fault tree constructed is only as good as the mental model conceived by the developer. To perform consistent diagnosis from fault trees, the trees must comprehensively represent the process causal relationships (explain all fault scenarios). There are no formal methods to verify the accuracy of the fault tree developed.

Fault trees have also been developed from digraphs (Lapp & Powers, 1977). Fault trees determine causal pathways through which primal events (faults) can propagate through the system to cause the top event (some significant malfunction). Given a top event, fault trees develop, using the process information in the form of digraphs, primal events or combinations of primal events which can result in the top event. The process information is provided in the form of unit models and the process topology showing the connectivity information among the units. Unit models represent how variables are related both when the unit is working and when it fails. The problem of fault-tree synthesis can be formulated as a search in finite state-space. Given an initial state, the algorithm applies 'operators' which transform the initial state to a goal state. Operators such as 'AND', 'OR' and 'XOR' are used in this transformation. The initial state is the top event. The goal state is the fault-tree connecting the top event to all possible primal events. Digraph defines the various possible states. Operators use the digraph to answer the question 'What could cause this?' at each node in the fault-tree. The answer to this is given by one of many alternatives in which case the node expands to these alternatives through an OR connection. If the answer involves all of a given set of alternatives, the node expands through an AND connection. Answering this question using only local digraph information is not possible when there are feedback and feedforward loops. Because of this, 'loop operators' are developed (Kelly & Lees, 1986; Lapp & Powers, 1977) which consider the function of the entire loop into picture when trying to explain 'what could cause this?'. Though fault trees might be generated in a rigorous fashion from digraphs, much of the representational appeal associated with fault trees is lost in this process.

Once a fault-tree is synthesized, the information from it is stored in the form of cut sets. A cut set is formally defined as any set of primal events which, when they occur simultaneously, cause the top event to occur. A minimal cut set is a subset of a cut set, which is a collection of all primal events that must occur together for the top event to occur. The collection of all minimal cut sets which can result in a top event specifies all the ways in which a given top event can happen. Fault trees do not serve the purpose of fault detection and cannot therefore be diagnostic systems. Ulerich and Powers (1988) derived a fault detection tree using the available real-time data to verify events in the fault tree. This is done by constructing an AND gate at each primal event. The inputs to this gate are the primal event and the real-time data that verify the event.

2.3. Qualitative physics

Qualitative physics or common sense reasoning about physical systems has been an area of major interest to the artificial intelligence community. Qualitative physics knowledge in fault diagnosis has been represented in mainly two ways. The first approach is to derive qualitative equations from the differential equations termed as confluence equations. The basic idea in the development of confluence equations is as follows. Consider the tank equations given in the digraph section. The steady-state confluence equations for this example are:

$$[F_2]-[F_1]=0$$

$$[Z]-[F_2]=0$$

where $[F_1]$ represents the sign of the deviation variable F_1. These equations can be solved qualitatively to get the qualitative values of the variables. An algebra can be specified which specifies how the qualitative values are combined. If the allowable value for a variable is $+$, then it means that the variable can take only qualitative high value, i.e., it can only increase.

An important thing to note here is that qualitative behavior can be derived even if an accurate mathematical model cannot be developed. Qualitative models do not require detailed information (such as exact expressions and numerical values) about the process. An order of magnitude information about the normal operating values of process parameters and variables is often sufficient. As an example, consider predicting level of the tank as the inlet flow rate increases. Using qualitative reasoning, one can predict that level would also increase (at least initially) without knowing the numerical values of cross sectional area of the tank, the outlet and so on.

Considerable work has been done in this area of qualitative modelling of systems and representation of causal knowledge. Simon (1977) and Iwasaki and Simon (1986) suggested the method of causal ordering which is used to get the causal relations by a suitable reduction of the functional relationships known about the process. The method known as causal ordering establishes, a priori to diagnosis, a partial or complete ordering between the variables in a system of equations. de Kleer and Brown (1984) suggested the use of confluence equations which can be developed from the differential equations in the mathematical model. They used the method of deviation propagation where they specify rules to establish an ordering but only at the time of propagating disturbances.

There is yet another method, called precedence ordering, that has been used to order the variables from the view point of information flow among them. The central idea is that the information flow among these equations is not simultaneous—a recognition of the presence of asymmetry (partial or complete precedence order among the variables) in the equations. This asymmetry shows the channels of information flow and thus represents causality. Precedence ordering has been studied widely for solving sets of algebraic equations simultaneously (Soylemez & Seider, 1973). Even though precedence ordering was developed for solving systems of equations, this technique can be used for deriving causal models as well. In fact, the concepts of causal ordering of Iwasaki and Simon (1986) and deviation propagation in confluence equations of de Kleer and Brown (1984) are related to this idea.

The other approach in qualitative physics is the derivation of qualitative behavior from the ordinary differential equations (ODEs). These qualitative behaviors for different failures can be used as a knowledge source. Sacks (1988) examines piece-wise linear approximations of nonlinear differential equations through the use of a qualitative mathematical reasoner to deduce the qualitative properties of the system. Kuipers (1986) predicts qualitative behavior by using qualitative differential equations (QDEs) that are an abstraction of the ODEs that represent the state of the system. The goals of these methodologies are to reason from qualitative physical and equational descriptions to qualitative behavioral descriptions and to provide explanations of behavior based on process observations and system description. The advantage of these qualitative simulators is their ability to yield partial conclusions from incomplete and often uncertain knowledge of the process. Each of the above theories start from a description of the physical mechanism, construct a model, and then use an algorithm so as to determine all of the behaviors of the system without precise knowledge of the parameters and functional relationships. de Kleer and Brown (1984) emphasize modelling individual physical components and deriving the behavior of a system of these components by using their connectivity to constrain the behavior of the overall system. Qualitative simulation as proposed by Kuipers involves specifying a constraint model of the physical process in terms of qualitative versions of mathematical relationships such as addition, multiplication, and differentiation. The variables used in modelling the physical system should satisfy these qualitative mathematical constraints. The resulting structure represents a qualitative abstraction of an ODE, or a QDE that models the process. In terms of applications of qualitative models in fault diagnosis, QSIM and qualitative process theory (QPT) have been the popular approaches and we review these approaches in some detail.

Conventionally, physical systems in science and engineering are modelled using differential equations, which are solved, either analytically or numerically to yield functions that represent the system behavior. Similarly, qualitative models represent an abstraction of the real physical system, and in terms of qualitative constraints, capture the information about the system. These qualitative models are 'solved' to get the qualitative behavioral description of the system. The QSIM representation and simulation algorithm allows us to reason mathematically about the description.

Qualitative simulation of a physical system by QSIM starts with a set of constraints modelling the structure of the process and its initial state and produces the envisionment—a graph consisting of all the possible future states of the system. Every path from the node to the root in the graph corresponds to a possible behavior of the system. The constraint model is a set of symbols representing the process variables, and a set of constraints on how these variables may be related to each other. The constraints allow us to express the simple mathematical relationships between the variables such as addition, multiplication and differentiation.

The fact that the variables in the qualitative simulation are continuously differentiable real-valued functions allows us to apply the mean value theorem, and restricts the possible transitions from a given qualitative description of state. The simulation starts with the initial state, generates all possible transitions that are allowed, and then employs the constraints to check which of the transitions are allowed by them. These transitions are then further filtered using global filters that detect whether a steady state has been reached, or a cyclic behavior is attained. Thus the successor state is obtained. If the possible successor states are more than one, the simulation branches, and a tree of qualitative behavioral descriptions is obtained.

A powerful feature of the QSIM algorithm is the ability to reason about the dynamic behavior of a system rather than just the steady state behaviour. To generate the behavioral description, the QSIM algorithm requires the structural description of the system in terms of the set of qualitative constraints, and the initial state of the system.

There are two main problems with deriving confluence equations from qualitative physics: ambiguities and spurious solutions. Ambiguities can be resolved completely only through the use of actual quantitative values. Frameworks for reasoning about relative orders of magnitudes have been proposed by Raiman (1986) and Mavrovouniotis and Stephanopoulos (1987). In these frameworks, influence magnitudes are related using relations such as A is negligible compared to B, A is close to B and A is the same order of magnitude as B. A set of inference rules then generates a partial ordering of values into groups significantly different in magnitude (Ungar & Venkatasubramanian, 1990). Spurious solutions refer to the generation of physically unrealizable solutions by a qualitative reasoning technique. This problem can be alleviated to a reasonable extent by modelling the system from different perspectives (Kay & Kuipers, 1993; Kuipers, 1985).

Ideas from QPT have also been used in process fault diagnosis (Grantham & Ungar, 1990, 1991). QPT construes physical systems as consisting of entities whose changes are caused by physical processes (Forbus, 1996). The domain is described by a collection of objects and each of these objects are defined completely by a qualitative state. The qualitative state is defined by a set of parameters which take on values in a quantity space. The state of the object is defined by the parameters represented by their position in the quantity space. The intervals in the quantity space are chosen to represent important events in the real number space. For example, the temperature space might consists of three intervals separated by its melting and boiling points. The relationships between variables are represented by qualitative proportionalities (Qprop). Qprop relations represent the response of the system to perturbations. To represent the primary cause of change, influence relations are used. Comparing Qprops and influences to traditional models, Qprops are analogous to algebraic equations and the influence relations are analogous to ordinary differential equations (Grantham & Ungar, 1990). These relationships are assumed to hold within what are known as individual views only. In general, views are representative of some physical phenomena. For example, a view could be describing the phenomena ‘two-component liquid’ (Grantham & Ungar, 1990).

Grantham and Ungar (1990) use the QPT framework to build a prototype first-principles troubleshooting system. Grantham and Ungar (1991) address the issue of building a comparative analysis system that accounts for the changes in physical system and modifies the underlying qualitative model. Further the system compares the original and modified models to analyze how structural changes affect behavior. The comparative analysis is again based on the QPT framework.

Another important contribution in the area of qualitative physics is the compositional modelling strategy for the development of tutoring and automatic model formulation in diagnostic systems. Compositional modelling is a strategy for organizing and reasoning about models of physical phenomenon that addresses the following problem: given an artifact description and a query, produce a model of the artifact that is commensurate with the needs of the query.

The key issue in compositional modelling is how to represent and organize knowledge about a domain so as to support automatic formulation of models. A domain theory that will represent knowledge would have to describe phenomenon at several levels of granularity, and incorporate multiple perspectives. Compositional modelling uses explicit modelling assumptions to decompose domain knowledge and semi-independent model fragments, each describing various aspects of objects and processes. Falkenhainer and Forbus (1991) layout a framework for compositional modelling. Given a general domain theory, a structural description of a specific system, and a query about system’s behavior, the model composition algorithm composes a model, which suffices to answer the query while minimizing extraneous detail.

2.4. Abstraction hierarchy of process knowledge

Another form of model knowledge is through the development of abstraction hierarchies based on decomposition. The idea of decomposition is to be able to draw inferences about the behavior of the overall system solely from the laws governing the behavior of its subsystems. In such a decomposition, the no-function-in-structure principle is central: the laws of the subsystem may not presume the functioning of the whole system (de Kleer & Brown, 1984). In a hierarchic description, one could represent a generic description of a class of process units. The governing equations describing an entire class of process units may make assumptions about the class as a whole but may not make any assumptions about the behavior of particular units. As an example, for a valve in pressure regulators, the area available for flow decreases as the pressure increases. However, this is not true for all valves and the general description of a class of valves may not assume this behavior. Another important principle for decomposition of systems is the principle of locality: the laws for a part specifically cannot refer to any other part. No-function-in-structure allows consistent behaviors among various units. Principle of locality permits the behavior to be predicted based only on local information. Popular decompositions of process systems are the following: (i) *structural:* specifies the connectivity information of a unit and (ii) *functional:* specifies the output of a unit as a function of its inputs (and possibly state information).

In abstraction hierarchies, the process system is decomposed into its process units. This decomposition allows a general representation of the functionality of a system in terms of the input–output relationships of its units. It is not important to the diagnostic system or the reasoning module that the functionality be expressed in qualitative or quantitative terms. One could consider other forms of process description to decompose the process system as well. For example, one can decompose the process system based on abstract functionalities. Moreover, there is no reason to restrict oneself to descriptions at the level of units. Decomposition of a process system to subsystems can be performed at various levels of abstraction. If the level of abstraction is control systems, then these subsystems represent various individual control loops (Finch & Kramer, 1987; Shafaghi, Androw, & Lees, 1984). If the level of abstraction is units, subsystems represent individual units. This brings us to the concept of abstraction hierarchies as the structural or functional description of a system.

There are two-dimensions along which abstraction at different levels is possible—structural and functional (Rasmussen, 1986). The structural hierarchy represents the connectivity information of the system and its subsystems. The functional abstraction hierarchy represents the means-end relationships between a system and its subsystems. Majority of the work in fault diagnosis in chemical engineering depends on the development of functional decomposition. Structural decomposition is an efficient decomposition in systems where there is a general equivalence between structure and function, like for example in an electrical circuit. The reason for the popularity of functional decomposition in chemical engineering is due to the complex functionalities of various units that cannot be expressed in terms of structure. Hence, the decomposition focused here is the functional decomposition. This functional hierarchy describes bottom–up, what various units with certain functions be used for and how they serve higher level purposes. They describe top–down how various purposes are implemented through various units with specific functions. The way the reasoning proceeds in a hierarchical description depends on the task at hand. Information on the proper function of a system is obtained from the levels above. Causes of improper function depend upon changes in the resources and limitations specified at lower levels and hence are explained bottom-up.

Diagnosis can be considered as a top-down search from a higher-level abstraction where groups of equipment and functional systems are considered to a lower-level of abstraction where individual units and unit functions are analyzed (Rasmussen, 1985). Based on this understanding, Shum and Davis (1985) decomposed the process into a hierarchy of functional subsystems. Each node in the hierarchy corresponds to the intended function of a subsystem. By comparing the function of the subsystem with the intended function, the hypothesis that a fault is present in that subsystem is evaluated. Finch and Kramer (1987) represent the plant as a set of interacting subsystems, where each subsystem is categorized as a control system (closed loops) or passive system (open loops) or an external system. Each of these subsystems has an associated function at this level of system description. Depending on their function, these subsystems are categorized as: (i) functional, stressed, uncontrolled or saturated in the case of control systems and (ii) functional or malfunctional in the case of passive and external systems.

These subsystems are described at a lower-level as process units, sensors, controllers, actuators and control elements. The idea is that the failure of the purpose of the higher-level subsystem is due to the failure of the function of one or more of these units and to use this higher level description to quickly identify the subsystem which is the source of malfunction. Functional search is a natural opening move in diagnosing complex process systems having subsystems with specific functions that are individually recognizable in the overall system response. This method has the virtue of avoiding unnecessary detail in the early stages of diagnosis and quickly focusing to the problem areas.

Hierarchical abstraction of mass and energy flow structures at different levels of function (called Multilevel Flow Models (MFM)), i.e., functional abstraction hierarchy, has been used by (Lind, 1991). An MFM model is a normative description of a system, a representation of its function in terms of what should be done, how should it be done and with what should it be done. This leads to three basic concepts in MFM: (i) goals, (ii) functions and (iii) physical components. Three types of connectional relations such as (i) achieve relations, (ii) condition relations and (iii) realize relations are used to connect objects. Diagnostic reasoning strategies based on the MFM model can be found in Larsson (1994). In this work, a functional representation for the process is provided in terms of the goals of the process, how these goals are achieved by a network of functions, how the functions depend on subgoals and how they are realized by physical components. The user chooses a goal for diagnosis and this could be at any functional level in terms of either the whole process or a part of the process. The search propagates downwards using achieve relationships checking if the functional goals of the subsystems are achieved. If a flow function conditioned by a subgoal is found to be at fault, then corresponding subgoal is investigated recursively. However, if the subgoal is working, then that part of the tree is skipped. This approach is demonstrated on a tank process by Larsson (1994). Walseth, Foss, Lind, and Ogaard (1992) show the utility of the MFM approach for a diagnostic application in a fertilizer plant. Recently, a hierarchical organization of diagnostic knowledge by primary processing systems, subsystems, components, behaviors and malfunction modes is described in Prasad, Davis, Jirapinyo, Josephson, and Bhalodia (1998). It is shown that such a hierarchical decomposition provides effective modularity for organizing large-scale diagnostic knowledge bases and also allows different techniques to be integrated to address specific local problems. Another work on hierarchical functional modelling can be found in Modarres and Cheon (1999).

3. Typology of diagnostic search strategies

There are fundamentally two different approaches to search in fault diagnosis (Rasmussen, 1986): topographic search, and symptomatic search. Topographic searches perform malfunction analysis using a template of normal operation, whereas, symptomatic searches look for symptoms to direct the search to the fault location. Fig. 2 shows a classification of diagnostic systems based on the search methods they employ.

3.1. Topographic search

Search can be performed in the mal-operating system with reference to a template representing normal or planned operation. The fault will be found as a mismatch and identified by its location in the system. This type of search is called topographic search.

3.1.1. Decomposition techniques

All topographic strategies depend on search with reference to a model of normal function and are therefore well suited for identification of disturbances that are not empirically known or that the designer has not foreseen. Consistency and correctness of the strategy does not depend on models of malfunction and hence is less influenced by multiple unknown disturbances. Since the faults are not known a priori, topological search helps only narrowing the focus of fault diagnosis to a subsystem. Fig. 4 shows how one can use this approach to check the functionality of various subsystems in a process.

The topographic search can be either a *functional* or *structural* search. In structural search, one first identifies the path of information flow from the input to the unit of interest and its output. If there is a fault in the path all the subcomponents participating in the information flow are included in the hypothesis set (Milne, 1987). Successive refinement is performed by selecting subpaths to localize the search. One performs this by collecting a set of good paths and bad paths. The subset of bad paths that also occur in the good paths is assumed to be correct and the cardinality of the hypothesis set is reduced.

In functional search, functionality of various groups of subcomponents is used to search for a fault. Refinements can be achieved by searching a hierarchy of submodels at various levels of detail. Similar to the structural search, a collection of normal submodels and abnormal submodels are identified. Individual components in both the normal and abnormal set is assumed to

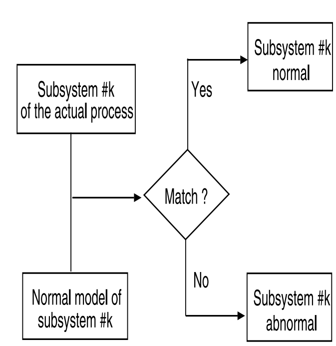

Fig. 4. Topographic search.

be normal thereby reducing the cardinality of the hypothesis set. In practice, though, a combination of structural and functional search is preferred.

The most important aspect of the topographic searches is in the lack of assumption about the faulty modes of operation. Assumption is made only about the normal operating mode. Hence these searches are relatively insensitive to novel and multiple faults and as such can be used to identify these malfunctions. An observation to note is that the search, more often than not, comes up with a set of hypotheses rather than the actual set of faults. Hence, though completeness might be guaranteed, the resolution of the fault set might be poor for practical utility.

3.2. Symptomatic search

A set of observations representing the abnormal state of the system can be used as a search template to find a matching set in a library of known symptoms related to different abnormal system conditions. This type of search is called symptomatic search. The main feature of these methods is that their decisions are derived from the structure of data sets, their internal relationships, and not from the topological structure of system properties. Symptomatic search is advantageous from the view point of information economy. Its limitation is that a reference pattern of the actual abnormal state of operation must be available, and multiple faults and novel disturbances may be difficult to identify.

In symptomatic approaches, the inputs to the process are also provided to the reference model. The outputs of the process are then compared to that of the reference model. When the parameters in the system are not directly measured, there are two different forms of identifying parameter variations between the process and the model. In a finite-state situation, with each state corresponding to a distinct partition of the parameter space, the process belongs to one of the finite possible states. In such cases if all the states are explicitly enumerated, one has the open loop formalism. If the states are generated and tested based on the mismatch feedback, one has the closed loop formalism.

3.2.1. Look-up tables

This is the simplest kind of symptomatic search. A template of abnormal behavior and corresponding symptoms are stored in the form of look-up tables. This kind of approach gets complicated and intractable for large-scale chemical systems. Hence one needs more systematic approaches for solving the diagnosis problems in the case of complicated plants.

3.2.2. Hypothesis and test search

Hypothesis and test search is a very popular symptomatic approach to diagnosis. If a search is based on reference patterns generated on-line by modification of a functional model, in correspondence with a hypothetical disturbance/fault, the search strategy is a hypothesis and test search in the closed-loop form. In this approach, hypotheses are generated sequentially and tested, as shown in Fig. 5. The efficiency of this search depends on the efficiency in generating hypotheses. Hypotheses are generated from topographic search or symptomatic search. In this approach, diagnosis proceeds in three steps: (a) hypothesis formulation; (b) determination of the effects of the hypothesized fault on the process (fault simulation); and (c) comparison of the result to plant data (hypothesis testing). If the predicted symptoms are wholly or partially present in the plant, the hypothesis may be retained, and the procedure repeated until no better hypothesis can be found. As the fault set would be very large, the set has to be reduced before fault simulations can be done. Compiled or heuristic knowledge is commonly used to reduce the fault set to give the hypothesis set.

Fig. 6 shows a schematic of the closed loop approach. In the closed loop approach, a candidate hypothesis is chosen and is tested if the reference matches the process. If the reference does not match the process, then a new reference is chosen for evaluation. In the closed loop approach, the information from the mismatch between the current reference and the process is used in the process of new hypothesis generation. In the open loop approach, many reference models are used with different hypotheses. By comparing reference and the process, the closest reference model is identified as the most representative of the process. Fig. 7 shows a schematic of the approach.

For hypothesis generation, a hypothesis graph G(P) for a closed diagnostic problem is evolved as the process of hypothesis generation and testing continues. Evolving the graph G(P) is essentially making explicit a portion of G(P) by a search procedure (Nilsson, 1980). The graph starts with a start node and a set of rules for generating successor nodes from any non-terminal node until a given termination condition is met. If the set of rules for generating successor nodes are heuristic, then the hypothesis generation through the evolution of the

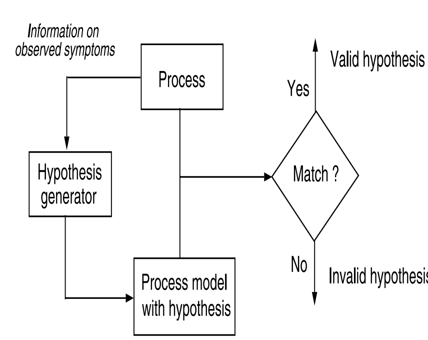

Fig. 5. Sequential hypothesis generation and testing.

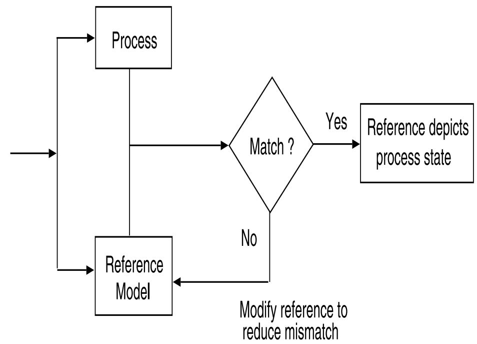

Fig. 6. Symptomatic search—closed loop approach.

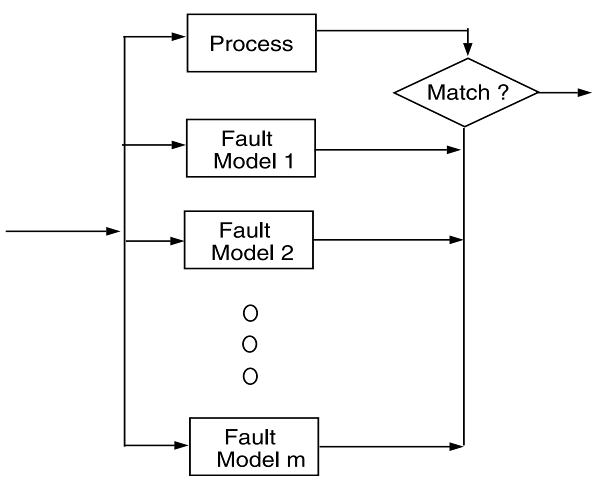

Fig. 7. Symptomatic search—open loop approach.

graph G(P) is also heuristic. The general state-space expansion of a graph is usually on the basis of 'costs' associated with the edges of the graph. An evaluation function decides, based on the costs, which nodes of the graph are to be opened. Peng and Reggia (1987a,b) developed a probabilistic method to overcome the combinatorial complexity in hypothesis-and-test approaches when multiple faults are considered. Their approach while limiting the number of hypotheses generated still guarantees the identification of the most likely hypotheses. This approach makes use of the 'parsimonious covering theory' or the 'Occum's razor' as the criterion for plausibility. Parsimonious covering theory postulates that a simple cover is preferable to a complex one. In other words, the simplest explanation is the most likely explanation for the cause of the system fault. Peng and Reggia (1987a,b) describe a probability based approach to guide the formation of hypotheses so that the most likely hypotheses are considered first.

One form of the open loop approach is the assumption-based approach exploiting redundancy. When a process fault occurs, the observed process behavior will be in conflict with the expected behavior of a normal process. This mismatch of behaviors can be attributed to the failure of certain assumptions about normality in the process. Each of the assumptions of normality assume the non-occurrence of a certain fault. The task of a diagnostic algorithm is then to identify the assumptions that result in the mismatch. Assumption-based methods require constraints to be developed based on distinct groups of assumptions. Once the constraints are obtained, they may be evaluated and checked for their satisfaction. The fault set is then identified based on the following method which assumes boolean logic for the expression of the satisfaction of a constraint. The satisfaction of the constraint can also be checked by using non-boolean logic as discussed by Petti, Klein, and Dhurjati (1990) in their Diagnostic Model Processor (DMP) algorithm. Another variant of the DMP approach which improves the resolution of the fault diagnostic algorithm using non-boolean constraint checking can be found in Chang, Yu, and Liou (1994, 1995).

A diagnosis is a conjecture that certain units are malfunctioning and the rest are functional. The problem is to specify which units we conjecture to be faulty. An assumption-based diagnostic algorithm searches for a diagnosis by making or dropping assumptions about a unit's function. Groups of assumptions together form invariant relationships about the process. These invariant relations can also be called constraints. Balance equations are an example of invariant relations. Assumptions in a balance equation may correspond to the various parameter values. In the description of a reactor, for example, an assumption may be made that the reactor is not leaking. The governing equations of a reactor would thus involve some assumptions about the expected function of the reactor. Furthermore, since the validity of a balance constraint can only be verified using sensor data, one should also consider the assumption of sensor accuracy. Validity of the assumptions guarantees the constraint (balance) satisfaction. Assumption-based methods assume that a set of constraints, each with a distinct set of assumptions, is available and that these constraints can be evaluated based on the sensor information from the process.

4. Conclusions

In this second part, of three parts, of review paper, various forms of qualitative models such as causal models and abstraction hierarchies were reviewed. Though qualitative models have a number of advantages as discussed in this paper, the major disadvantage is the generation of spurious solutions. Considerable amount of work has been done in the reduction of the number of spurious solutions while reasoning with qualitative models. In SDGs, this is done using generation of latent constraints and similar techniques have been proposed for qualitative physics based models such as QSIM. The search strategies were classified as either topographic or symptomatic search and the difference between these two types of search strategies were highlighted. Clearly, for a given qualitative representation, different search strategies could be used for diagnosis. Hence, one can view the methods proposed in the literature as different combinations of the qualitative methods and search strategies reviewed in this paper.

References

Bennetts, R. G. (1974). Comment on the evaluation of fault trees. *IEEE Transactions on Reliability 23* (1), 54–55.

Chang, C. C., & Yu, C. C. (1990). On-line fault diagnosis using the signed directed graph. *Industrial and Engineering Chemistry Research 29* (7), 1290–1299.

Chang, I., Yu, C., & Liou, C. (1994). Model-based approach for fault diagnosis I. Principles of deep model algorithm. *Industrial and Engineering Chemistry Research 33* (6), 1542–1555.

Chang, I., Yu, C., & Liou, C. (1995). Model-based approach for fault diagnosis II. Extension to interval systems. *Industrial and Engineering Chemistry Research 34* (3), 828–844.

Charniak, E., & McDermott, D. (1984). *Introduction to artificial intelligence*. Massechusetts: Addison-Wesley Publishing Company.

Davis, R. (1984). Diagnosis reasoning based on structure and behavior. *Artificial Intelligence 24* (1–3), 347–410.

de Kleer, J., & Brown, S. (1984). A qualitative physics based on confluences. *Artificial Intelligence 24* (1–3), 7–83.

Falkenhainer, B., & Forbus, K. (1991). Compositional modeling: finding the right model for the job. *Artificial Intelligence 51*, 95–143.

Finch, F. E., & Kramer, M. A. (1987). Narrowing diagnostic focus using functional decomposition. *American Institute of Chemical Engineers Journal 34* (1), 130–140.

Forbus, K. (1996). *Qualitative reasoning*. CRC Handbook of Computer Science.

Fussell, J. B. (1974). Fault tree analysis—state of the art. *IEEE Transactions on Reliability 23* (1), 51–53.

Genovesi, A., Harmand, J., & Steyer, J. P. (1999). A fuzzy logic based diagnosis system for the on-line supervision of an anaerobic digestor pilot-plant. *Biochemical Engineering Journal 3* (3), 171–183.

Grantham, S. D., & Ungar, L. H. (1990). A first principles approach to automated troubleshooting of chemical plants. *Computers and Chemical Engineering 14* (7), 783–798.

Grantham, S. D., & Ungar, L. H. (1991). Comparative analysis of qualitative models when the model changes. *American Institute of Chemical Engineers Journal 37* (6), 931–943.

Han, C., Shih, R., & Lee, L. (1994). Quantifying signed directed graphs with the fuzzy set for fault diagnosis resolution improvement. *Industrial and Engineering Chemistry Research 33* (8), 1943–1954.

Iri, M., Aoki, K., O'Shima, E., & Matsuyama, H. (1979). An algorithm for diagnosis of system failures in the chemical process. *Computers and Chemical Engineering 3* (1–4), 489–493.

Iwasaki, Y., & Simon, H. A. (1986). Causality in device behavior. *Artificial Intelligence 29* (1), 3–32.

Kay, H., Kuipers, B. (1993). Numerical behavior envelopes for qualitative models. *Proceedings of AAAI-93* , Menlo Park, CA, USA (pp. 606–613).

Kelly, B. E., & Lees, F. P. (1986). The propagation of faults in process plants: 2. fault tree synthesis. *Reliability Engineering 16* (1), 39–62.

Kim, H., & Lee, K. (1998). Fuzzy implications of fuzzy cognitive map with emphasis on fuzzy causal relationship and fuzzy partially causal relationship. *Fuzzy Sets and Systems 97* (3), 303–313.

Kokawa, M., Satoshi, M., & Shigai, S. (1983). Fault location using digraph and inverse direction search with application. *Automatica 19* (6), 729–735.

Kramer, M. A., & Palowitch, B. L. (1987). A rule based approach to fault diagnosis using the signed directed graph. *American Institute of Chemical Engineers Journal 33* (7), 1067–1078.

Kuipers, B. (1985). The limits of qualitative simulation. In *Proceedings of ninth joint international conference on artificial intelligence*.

Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence 29* (3), 289–338.

Lapp, S. A., & Powers, G. A. (1977). Computer-aided synthesis of fault trees. *IEEE Transactions on Reliability 26* (1), 2–13.

Larsson, J. E. (1994). Diagnostic reasoning strategies for means-end models. *Automatica 30* (5), 775–787.

Li, R., & Wang, X. (2001). Qualitative/quantitative simulation of process temporal behavior using clustered fuzzy digraphs. *American Institute of Chemical Engineers Journal 47* (4), 906–919.

Lind, M. (1991) Abstraction for modeling diagnostic strategies. In *IFAC workshop on computer software structures integrating AI/KBS systems in process control*.

Mavrovouniotis, M., Stephanopoulos, G. (1987). Reasoning with order of magnitudes and approximate relations. *Proceedings of AAAI-87*, *July*.

Milne, R. (1987). Strategies for diagnosis. *IEEE Transactions on Systems, Man and Cybernetics 17* (3), 333–339.

Modarres, M., & Cheon, S. W. (1999). Function-centered modeling of engineering systems using the goal tree-success tree technique and functional primitives. *Reliability Engineering and System Safety 64* (2), 181–200.

Nilsson, N. (1980). *Principles of artificial intelligence*. Los Altos, CA: Morgan Kaufmann.

Oyeleye, O. O., & Kramer, M. A. (1988). Qualitative simulation of chemical process systems: steady state analysis. *American Institute of Chemical Engineers Journal 34* (9), 1441–1454.

Peng, Y., & Reggia, J. A. (1987a). A probabilistic causal model for diagnostic problem solving—Part I: integrating symbolic causal inference with numeric probabilistic inference. *IEEE Transactions on Systems, Man and Cybernetics 17* (2), 146–162.

Peng, Y., & Reggia, J. A. (1987b). A probabilistic causal model for diagnostic problem solving—Part II: diagnostic search. *IEEE Transactions on Systems, Man and Cybernetics 17* (3), 395–406.

Petti, T. F., Klein, J., & Dhurjati, P. S. (1990). Diagnostic model processor: using deep knowledge for process fault diagnosis. *American Institute of Chemical Engineers Journal 36* (4), 565–575.

Prasad, P. R., Davis, J. F., Jirapinyo, Y., Josephson, J. R., & Bhalodia, M. (1998). Structuring diagnostic knowledge for large-scale process systems. *Computers and Chemical Engineering 22* (12), 1897–1905.

Raiman, O. (1986). Order of magnitude reasoning. In *Proceedings of AAAI-86, pp 100–104, August*.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man and Cybernetics 15* (2), 234–243.

Rasmussen, J. (1986). *Information processing and human–machine interaction*. New York: North Holland.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence 32* (1), 57–95.

Rich, S. H., & Venkatasubramanian, V. (1987). Model-based reasoning in diagnostic expert systems for chemical process plants. *Computers and Chemical Engineering 11* (2), 111–122.

Rich, S. H., & Venkatasubramanian, V. (1989). Causality-based failure-driven learning in diagnostic expert systems. *American Institute of Chemical Engineers Journal 35* (6), 943–950.

Sacks, E. (1988). Qualitative analysis of piecewise linear approximation. *Journal of Artificial Intelligence in Engineering 3* (3), 151–155.

Shafaghi, A., Androw, P. K., & Lees, F. P. (1984). Fault tree synthesis based on control loop structure. *Chemical Engineering Research and Design 62* (2), 101–110.

Shih, R., & Lee, L. (1995a). Use of fuzzy cause–effect digraph for resolution fault diagnosis for process plants I. Fuzzy cause–effect digraph. *Industrial and Engineering Chemistry Research 34* (5), 1688–1702.

Shih, R., & Lee, L. (1995b). Use of fuzzy cause–effect digraph for resolution fault diagnosis for process plants II. Diagnostic algorithm and applications. *Industrial and Engineering Chemistry Research 34* (5), 1703–1717.

Shiozaki, J., Matsuyama, H., O'Shima, E., & Iri, M. (1985). An improved algorithm for diagnosis of system failures in the chemical process. *Computers and Chemical Engineering 9* (3), 285–293.

Shiozaki, J., Matsuyama, H., Tano, K., & O'Shima, E. (1985). Fault diagnosis of chemical processes by the use of signed, directed graphs. Extension to five-range patterns of abnormality. *International Chemical Engineering 25* (4), 651–659.

Shum, S.K., Davis, J.F. (1985). An expert system for diagnosing process plant malfunctions. In *IFAC workshop on fault detection and safety in chemical plants*, Kyoto, Japan.

Simon, H. A. (1977). *Models of discovery*. Boston: Reidel Publishing Company.

Soylemez, S., & Seider, W. D. (1973). A new technique for precedence-ordering chemical process equation sets. *American Institute of Chemical Engineers Journal 19* (5), 934–942.

Tarifa, E., & Scenna, N. (1997). Fault diagnosis, directed graphs, and fuzzy logic. *Computers and Chemical Engineering 21*, S649–S654.

Ulerich, N. H., & Powers, G. A. (1988). Online hazard aversion and fault diagnosis in chemical processes: the digraph+fault tree method. *IEEE Transactions on Reliability 37* (2), 171–177.

Umeda, T., Kuriyama, T., O'shima, E., & Matsuyama, H. (1980). A graphical approach to cause and effect analysis of chemical processing systems. *Chemical Engineering Science 35* (12), 2379–2388.

Ungar, L. H., & Venkatasubramanian, V. (1990). *Artificial intelligence in process systems engineering, Vol III: knowledge representation*. Austin,Texas: CACHE.

Vaidhyanathan, R., & Venkatasubramanian, V. (1995). Digraph-based models for automated HAZOP analysis. *Reliability Engineering and Systems Safety 50* (1), 33–49.

Vedam, H., & Venkatasubramanian, V. (1997). Signed digraph based multiple fault diagnosis. *Computers and Chemical Engineering 21*, S655–S660.

Venkatasubramanian, V., & Rich, S. H. (1988). An object-oriented two-tier architecture for integrating compiled and deep-level knowledge for process diagnosis. *Computers and Chemical Engineering 12* (9–10), 903–921.

Walseth, J., Foss, B.A., Lind, M., Ogaard, O. (1992). Models for diagnosis—application to a fertilizer plant. In *IFAC symposium on online fault detection and supervision in the chemical process industries*, 197–202.

Wang, X., Yang, S., Veloso, E., Lu, M., & McGreavy, C. (1995). Qualitative process modeling—a fuzzy signed directed graph method. *Computers and Chemical Engineering 19*, S735–S740.

Wilcox, N. A., & Himmelblau, D. M. (1994a). Possible cause and effect graphs (PCEG) model for fault diagnosis I. Methodology. *Computers and Chemical Engineering 18* (2), 103–116.

Wilcox, N. A., & Himmelblau, D. M. (1994b). Possible cause and effect graphs (PCEG) model for fault diagnosis II. Applications. *Computers and Chemical Engineering 18* (2), 117–127.