3

Qualitative Fault Detection and Hazard Analysis Based on Signed Directed Graphs for Large-Scale Complex Systems

Fan Yang[1,2], Deyun Xiao[1] and Sirish L. Shah[2]
[1] Tsinghua University
[2] University of Alberta
[1] China
[2] Canada

1. Introduction

Nowadays in modern industries, the scale and complexity of process systems are increased continuously. These systems are subject to low productivity, system faults or even hazards because of various conditions such as mis-operation, equipment quality change, external disturbance, and control system failure. In these systems, many elements are interacted, so a local fault can be propagated and probably spread to a wide range. Thus it is of great importance to find the possible root causes and consequences according to the current symptom promptly. Compared with the classic fault detection for local systems, the fault detection for large-scale complex systems concerns more about the fault propagation in the overall systems. And this demand is much close to hazard analysis for the system risks, which is a kind of qualitative analysis in most cases prior to quantitative analysis.
The signed directed graph (SDG) model is a kind of qualitative graphical models to describe the process variables and their cause-effect relations in continuous systems, denoting the process variables as nodes while causal relations as directed arcs. The signs of nodes and arc correspond to variable deviations and causal directions individually. The SDG obtained by flowsheets, empirical knowledge and mathematical models is an expression of deep knowledge. Based on the graph search, fault propagation paths can be obtained and thus certainly be helpful for the analysis of root causes and sequences (Yang & Xiao, 2005a). And with development of the computer-aided technology, graph theory has been implemented successfully by some graph editors, some of which, like Graphviz (2009), can transform text description into graphs easily. Hence the SDG technology can be easily combined with the other design, analysis and management tools.
The SDG definition and its application in fault diagnosis were firstly presented by Iri et al. (1979). Ever since then, many scholars have contributed to this area, including modeling, inference, software development and applications. Many efforts have been particularly made to implement the methods and to overcome the disadvantages, such as spurious solutions. Here we recognize some representatives among them. Kramer & Palowitch (1987)

used rules to describe SDG arcs, which shows that expert systems can be employed as a tool in this problem. Oyeleye & Kramer (1988) took into account the qualitative simulation for the SDG inference. Shiozaki et al. (1989) improved the SDG model by adding fault revealing time. Yu et al. (Chang & Yu, 1990; Yu & Lee, 1991) introduced fuzzy information for arc signs to describe the steady state gains. Maurya et al. (2003a, 2003b, 2006) described the modeling method based on differential equations (DEs) and algebraic equations (AEs), analyzed the initial and final responses based on SDGs, and studied the description and analysis of control loops. SDG method has been combined with other data-driven methods to improve the diagnosis accuracy (Vedam & Venkatasubramanian, 1999; Lee et al., 2006). At first, the inference is based on single fault assumption, but multiple fault cases attract more and more attention (Vedam & Venkatasubramanian, 1997; Zhang et al. 2005; Chen & Chang, 2007). Up to now, SDG method has been implemented in some software tools (Mylaraswamy & Venkatasubramanian, 1997; McCoy et al. 1999; Zhang et al., 2005) and applied in various industrial systems.

Aiming at SDG applications in the area of fault detection and hazard analysis, the problems of description and inference are most important. As the system extends, the time consumption of graph search is heavy, so the single-level SDG model should be transformed into hierarchical model to improve the search efficiency. The root cause can be searched in this model level by level according to the initial response of the system. In control systems and many other cases, cycles exist in the graph, resulting in the truncation or misleading to the search. Thus the theoretic fundamentals and dynamic features of SDGs should be studied. We have analyzed the fault propagation principles by operations of corresponding qualitative matrices and obtained some typical rules of control systems.

Moreover, fault detection is performed based on sensor readings, thus the sensor location strategy affects the performance of fault detection. Due to the economical or technical limitations, the number of sensors should be limited while meeting the demands of fault detection. This can be considered in the SDG framework. We analyze main criteria such as detectability, identifiability and reliability in the framework of SDGs and presented algorithms, in order to guarantee that the faults can be detected and identified, and to optimize the fault detection ability.

This chapter is organized as follows: first, the SDG description is reviewed and hierarchical model is indicated; then the fault propagation rules and inference approaches are summarized to lead to the successful application of fault detection and hazard analysis; some considerations about sensor location are introduced next; finally a generator set process in a power plant is modeled and analyzed to illustrate the proposed model and method.

2. Model Description of Signed Directed Graph

2.1 Basic Form of SDG Model

SDGs are established by representing the process variables as graph nodes and representing causal relations as directed arcs. An arc from node *A* to node *B* implies that the deviation of *A* may cause the deviation of *B*. For convenience, "+", "-" or "0" is assigned to the nodes in comparison with normal operating value thresholds to denote higher than, lower than or within the normal region respectively. Positive or negative influence between nodes is distinguished by the sign "+" (promotion) or "-" (suppression), assigned to the arc (Iri et al., 1979). The definition is as follows:

Definition 1: An *SDG model* γ is the composite (G,φ) of (1) a digraph G which is the quadruple $(N, A, \partial^+, \partial^-)$ of (a) a set of nodes $N=\{v_1, v_2, \cdots, v_n\}$, (b) a set of arcs $A=\{a_1, a_2, \cdots, a_m\}$, (c) a couple of incidence relations $\partial^+ : A \to N$ and $\partial^- : A \to N$ which make each arc correspond to its initial node $\partial^+ a_k$ and its terminal node $\partial^- a_k$, respectively; and (2) a function $\varphi : A \to \{+,-\}$, where $\varphi(a_k)$ $(a_k \in A)$ is called *the sign of arc* a_k.

Usually we use a_{ij} to denote the arc from v_i to v_j.

Definition 2: A *pattern* on the SDG model $\gamma = (G,\varphi)$ is a function ψ: $N \to \{+,0,-\}$. $\psi(v)$ $(v \in N)$ is called the *sign of node* v, i.e.

$$\psi(v) = 0 \text{ for } |x_v - \bar{x}_v| < \varepsilon_v$$
$$\psi(v) = + \text{ for } x_v - \bar{x}_v \ge \varepsilon_v$$
$$\psi(v) = - \text{ for } \bar{x}_v - x_v \ge \varepsilon_v$$

where x_v is the measurement of the variable v, $\bar{x}_v$ is the normal value, and ε_v is the threshold.

Definition 3: Given a pattern ψ on a SDG model $\gamma = (G,\varphi)$, an arc a is said to be *consistent* (with ψ) if $\psi(\partial^+ a)\varphi(a)\psi(\partial^- a) = +$. A path, which is consisted of arcs $a_1, a_2, \cdots, a_k$ linked successively, is said to be *consistent* (with ψ) if $\psi(\partial^+ a_1)\varphi(a_1)\cdots\varphi(a_k)\psi(\partial^- a_k) = +$.

2.2 Modeling Methods of SDGs

2.2.1 SDG modeling by mathematical equations

In general, SDGs can be obtained either from operational data and process knowledge, or mathematical models. If we have the differential algebraic equations (DAEs), then we can derive the structure and signs of the graph by specific methods (Maurya, 2003a).

A typical dynamic system can be expressed as a set of DEs

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = f_i\left(x_1, \cdots, x_n\right) \tag{1}$$

where $x_1, \cdots, x_n$ are state variables. By Taylor expansion near normal state, we get

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} \approx f_i\left(x_1^0, \cdots, x_n^0\right) + \sum_{j=1}^{n} \left.\frac{\partial f_i}{\partial x_j}\right|_{x_1^0, \cdots, x_n^0} \left(x_j - x_j^0\right) \tag{2}$$

where $x_1^0, \cdots, x_n^0$ are normal states. Eq. (2) can be written as the following matrix form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \approx \begin{bmatrix} f_1\left(x_1^0,\cdots,x_n^0\right) \\ \vdots \\ f_n\left(x_1^0,\cdots,x_n^0\right) \end{bmatrix} + \left.\begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{x_1^0,\cdots,x_n^0} \begin{bmatrix} x_1 - x_1^0 \\ \vdots \\ x_n - x_n^0 \end{bmatrix} \tag{3}$$

The Jacobian matrix

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{4}$$

can be described by an SDG whose signs of arcs are defined as

$$\operatorname{sgn}\left(x_j \to x_i\right) = \operatorname{sgn}\left[\left.\frac{\partial f_i}{\partial x_j}\right|_{x_1^0,\cdots,x_n^0}\right] \tag{5}$$

if the nodes correspond to the state variables. Thus the SDG actually describes the direct influences or sensitivities between state variables.

In practical problems, the systems often have the following form as DEs:

$$a_n\left(\mathrm{d}^n x/\mathrm{d}t^n\right) + \cdots + a_2\left(\mathrm{d}^2 x/\mathrm{d}t^2\right) + a_1\left(\mathrm{d}x/\mathrm{d}t\right) + a_0 x = e \tag{6}$$

where x is the state and e is the disturbance. When $n=1$, it is a first-order system:

$$\left(\mathrm{d}/\mathrm{d}t\right)x = -\left(a_0/a_1\right)x + \left(1/a_1\right)e \tag{7}$$

The step response is shown as Fig. 1(a). An arc is constructed from the node e to x with a sign sgn[$1/a_1$] and a self-cycle on the node x with a sign -sgn[a_0/a_1]. For high-order systems, simplification can be made because the corresponding DE includes different order derivatives of the same variable, which can be avoided for the explicit physical meaning of the model. They can be approximated as first-order systems with delays:

$$\left(\mathrm{d}/\mathrm{d}t\right)x(t-\tau) = -\left(a_0'/a_1'\right)x(t) + \left(1/a_1'\right)e(t) \tag{8}$$

where τ is the equivalent pure delay. Its step response is shown as Fig. 1(b). The method of constructing SDGs is the same as the former one, and the delay can be embodied in dynamic SDGs (Yang & Xiao, 2006a).

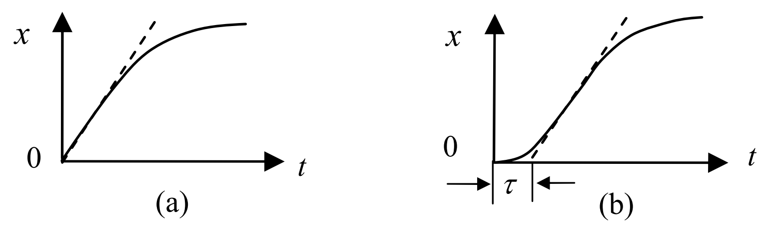

Fig. 1. Step response of different systems. (a) First-order system, (b) High-order system

Algebraic equations are usually included in the mathematical models as constraints which can also be transformed into SDGs (Maurya et al., 2003a) although they are noncausal in nature. Because there may be multiple perfect matchings between equations and variables, the corresponding SDGs may not be unique. Some treatment should be made to screen the unsteady or spurious SDGs (Oyeleye & Kramer, 1988; Maurya et al., 2003a).
For example, a tank system is shown as Fig. 2(a) where L is the level in the tank, R is resistance in the outlet pipe (can be manipulated by a valve), F_1 and F_2 are inlet and outlet flowrates respectively. The system is described as following DAEs:

$$F_2 = \frac{\alpha}{R}\sqrt{L} \tag{9}$$

$$A\frac{\mathrm{d}}{\mathrm{d}t}L = F_1 - F_2 \tag{10}$$

where A is the cross sectional area of the tank, and α is a constant. By the above method, the SDG is set up as Fig. 2(b).

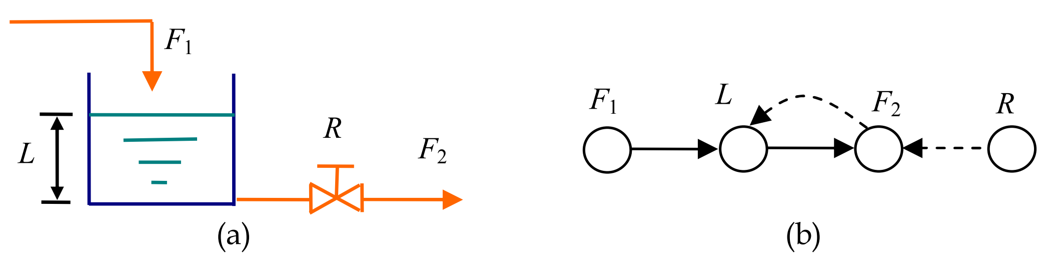

Fig. 2. Tank system and its SDG. (a) Schematic, (b) SDG

2.2.2 SDG modeling by qualitative process knowledge

In more cases, the SDG is established by qualitative process knowledge and experience. Fig. 3(a) shows a tank with one inlet and two outlets with control. The arcs from F_2 to V_2 and L to V_3 in Fig. 3(b) describe the flowrate control and level control respectively. Each control loop can be expressed by a negative cycle in SDG because of the negative feedback action. This qualitative SDG can be obtained directly from process knowledge and does need the exact mathematical equations. Sometimes the qualitative simulation and sensitivity experiments may also help. The SDGs obtained by this method often include indirect causalities besides direct ones, so the graph should be simplified and transformed so that all the arcs stand for direct causalities. Some rules are summarized by Yang & Xiao (2005b).

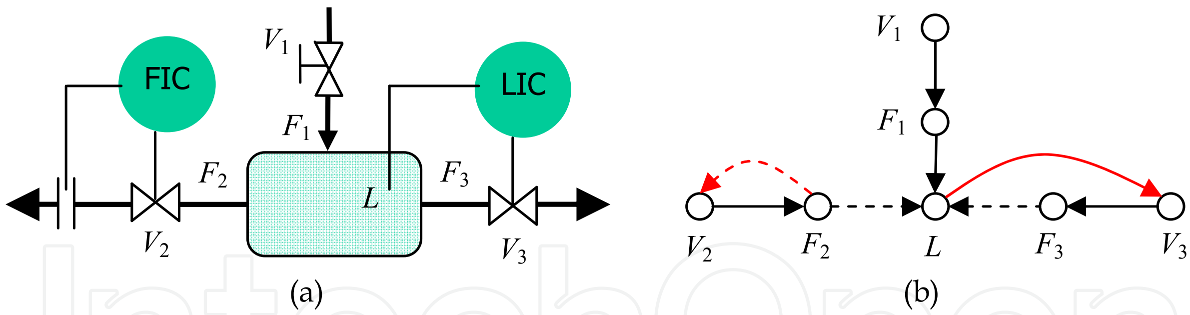

Fig. 3. Schematic and SDG of tank system with controlled flowrates. (a) Schematic, (b) SDG

Besides, P&ID diagrams and other flowsheets are very important topological process knowledge expression that can be standardized in XML (extensible markup language) format. It has been implemented in some commercial software products such as SmartPlant P&ID from Intergraph. The topology or connectivity obtained here includes both material flow and information flow, which are needed for SDG modeling. Although the granularity is entity-based, which is not enough for the variable-based SDG modeling, this kind of topological information is the fundamental of SDG and can be used as references as well (Thambirajah et al., 2009).

The SDG set up by the above methods can be validated by process data. For example, correlation is a necessary condition of causality, so the cross-correlation between every two measured variables can be used to validate the arcs in SDGs, and the directions can also be obtained by shifting the time series to find the maximal cross-correlation. Alternatively, probabilistic measure such as transfer entropy can be used to obtain the causality and directionality (Bauer et al., 2007).

In summary, the main steps of SDG modeling are: (1) Collect process knowledge, especially P&ID diagram and equations. (2) Set up the material flow diagraph by connectivity information between entities. (3) Choose the key variables and give them signs according to the process knowledge. (4) Add control arcs on the diagraph to constitute the SDG skeleton. (5) Add other variables and arcs to form the entire SDG. (6) Simplify and verify the SDG by graph theory. (7) Validate the SDG with process data and sensitivity experiments.

2.3 Hierarchical SDG Description of Large-Scale Complex Systems

Based on the decomposition-aggregation approach, a single-level SDG model can be transformed into a hierarchical model (Gentil & Montmain, 2004; Preisig, 2009). With this model, it is clear and easy to understand the system inherently. As such, the fault analysis method should also be modified from a centralized one to a distributed one.

The whole SDG model can be classified into 3 levels. If the scale of the whole system is too large, then more levels can be established, but 3-level model is enough for most cases. So we take it as a typical pyramid structure. The top level is called system level, where the system is divided into several sub-systems. Sometimes a large-scale system may include several independent sub-systems which can be dealt with separately. Also, in many cases, several components are operated in sequence or in parallel, with no recycle or other kind of interactions existed across the different components, then these components can also be regarded as sub-systems. Of course, if the SDG of the whole system is connected and cannot be separated, then it composes the only sub-system itself.

In the middle level, each control system is regarded as a super-node and the relations between control systems are expressed by arcs among controlled variables and a few important manipulated variables or other variables. The signs of arcs are determined according to the propagation rules to assure the consistency of the paths. The variables in some control loop and not appeared in other part of the system are usually invisible here. The SDG in this level is the backbone of the system which shows the main connectivity in the system flowsheet.

The bottom-level SDGs are the SDG units of all the control systems. The description is the most detailed qualitative expression because it shows the causalities between variables. Since most of the control systems are based on feedback actions, each SDG in this level usually contains at least a loop with various bias nodes attached on them.

2.4 Matrix Explanation of SDG Model

In this section, we look at the SDG model from another viewpoint. An SDG can be also described as an adjacency matrix $\boldsymbol{X}$ with the element 1/0 denoting the direct adjacency and direction between two variables. Actually it is the transpose of Jaccobian matrix in Eq. (4) with unsigned elements. By matrix computations, reachability matrix $\boldsymbol{R}$ can be obtained from $\boldsymbol{X}$, which shows the directed reachability from one variable to another, in which the element 1 means there are at least a path in the corresponding SDG (Jiang et al., 2008). It can be observed that the computation is just another form of graph traversal.

By simultaneous permutation of row and column (with variable order changed), $\boldsymbol{X}$ can be block triangulated as follows:

$$\boldsymbol{X}' = \boldsymbol{TXT}^T = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1m} \\ 0 & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{A}_{mm} \end{bmatrix} \tag{11}$$

Each block in the diagonal denotes a sub-system with a partial order meaning that the sub-system with larger number can not reach the one with smaller number. It can also be explained by the reachability matrix which is definitely also block triangulated with the same order as:

$$\boldsymbol{R} = \left(\boldsymbol{X}' + \boldsymbol{X}'^2 + \cdots + \boldsymbol{X}'^n\right)^{\#} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} & \cdots & \boldsymbol{B}_{1m} \\ 0 & \boldsymbol{B}_{22} & \cdots & \boldsymbol{B}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{B}_{mm} \end{bmatrix} \tag{12}$$

where the sign # is the Boolean equivalent (Mah, 1989). If the intersection block $\boldsymbol{B}_{ij}$ is a zero matrix, then the corresponding two sub-systems are independent (no arcs between them), otherwise they are in sequence. Thus we explain the decomposition between the top and middle level.

When we look at the relationship among control systems, we take a control loop as a super-node and add an arc from node i to node j, if the controller output of controller i can directly

affect the controlled variable of controller *j* without going through controller output of any other nodes. This SDG as a part of the middle-level SDG is also named as control loop diagraph (Jiang, 2008).
For a feedback control system, there exists a loop in the corresponding SDG. Thus according to the controllability concept, all the variables within the loop are strongly connected, which can be found in the reachability matrix as a block with all the elements are ones.
Let us look at the tank example as Fig. 3 and get the adjacency matrix and reachability matrix by Eq. (12) as follows, both of which are block triangulated.

$$\boldsymbol{X}=\left[\begin{array}{cc:cc:ccc} 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1\\ \hdashline 0&0&0&1&0&0&0\\ 0&0&1&0&0&0&1\\ \hdashline 0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&1&0&0 \end{array}\right],\ \boldsymbol{R}=\left[\begin{array}{cc:cc:ccc} 0&1&0&0&1&1&1\\ 0&0&0&0&1&1&1\\ \hdashline 0&0&1&1&1&1&1\\ 0&0&1&1&1&1&1\\ \hdashline 0&0&0&0&1&1&1\\ 0&0&0&0&1&1&1\\ 0&0&0&0&1&1&1 \end{array}\right] \tag{13}$$

where the variable order is V_1, F_1, V_2, F_2, V_3, F_3, L. They are divided into 3 groups: inlet (V_1 and F_1), one outlet with flowrate control (V_2 and F_2) and another outlet with level control (V_3, F_3 and L). The elements of $\boldsymbol{R}_{22}$ and $\boldsymbol{R}_{33}$ are all ones because they are control loops, and the elements of $\boldsymbol{R}_{23}$ are ones showing the flowrate controller influences the level controller. Hence the control loop diagraph is consisted of two nodes corresponding to the two controllers and an arc corresponding to the influence between them. Moreover, if the variable order is changed to put V_1 and V_2 meaning the two controller outputs at the last, the corresponding block is just the adjacency matrix of the control loop diagraph. This is a useful property that links the concepts of SDG, control loop digraph and the matrices.
Matrix explanation helps us understand the SDG concept and its potential in applications. In fact, some results, such as propagation rules, are derived from matrix description.

3. Inference Approaches Based on SDGs

3.1 Fault Propagation Rules

Based on the SDG description, the fault propagation can be described qualitatively. There are two basic principles:
Proposition 1: The fault is propagated along the consistent paths.
Proposition 2: The node signs are determined by nodal balance, i.e. the sign on each node must be equal to the net influence on the node:

$$\psi(v_j)=\sum_i \varphi(a_{ij})\cdot\psi(v_i) \tag{14}$$

where the qualitative operation rules are as Table 1. Due to the loss of quantitative information, some signs can not be determined shown as '?' in the table, which causes the uncertainty in the solutions.

No.	sgn[x]	sgn[y]	sgn[x]+sgn[y]	sgn[x]·sgn[y]
1	0	sgn[y]	sgn[y]	0
2	±	sgn[x]	sgn[x]	+
3	±	-sgn[x]	?	-

Table 1. Qualitative operation rules

The logic on a node in SDGs is OR in nature because any input deviation can result in the node sign. In some cases, however, there are other types of logics, for example, the logic is AND, XOR or high/low-selective, or arcs or nodes are conditional, some necessary logic nodes should be added to the SDG (Yang & Xiao, 2007).

Proposition 1 can be easily understood. By testing the consistency one can find the fault propagation paths based on the measurements, which form a sub-graph of the original SDG, called cause-effect graph (Iri et al., 1979). On the other hand, one can predict the next step response based on the measured and assumed variables.

Proposition 2, however, may have some limitations because it is only suitable for the dynamic trends near the initial state. When a fault occurs, the response of variables can be divided into three stages – initial, intermediate, and final responses. In large-scale complex systems, the intermediate response is very complex, but in most cases, we concentrate only on the initial and final stages. For stable systems with fixed input, the final response is a steady state. Thus the input and exogenous disturbances are assumed as step functions to show abrupt changes.

Initial response is the first response just after the exogenous input changes. In dynamic systems expressed by DAEs, initial response is the nonzero response of system variables predicted by propagation through all the shortest paths in the corresponding SDG if we define the length of arcs in AE and DE portion by 0 and 1 respectively (Maurya, 2003a). Final response is the steady states of variables obtained after the dynamic period ends. It can be solved simply by setting the derivatives as zeros in DE portion of DAE. For the obtained AEs, the final response can be predicted by propagation through all the directed (acyclic) paths in the corresponding SDG. However, there may exist more than one perfect matching between equations and variables, thus there may exist more than one SDG corresponding to the AEs. If there is only one perfect matching, the above method is correct; otherwise, the result may be wrong because the results based on different perfect matchings are inconsistent. There is an exception, however, if an SDG corresponding to a perfect matching contains only negative cycles, then any perfect matching (for which the SDG contains only negative cycles) can be chosen and the final response can be decided using the above method (Maurya, 2003a).

3.2 Control Action Influences on Fault Propagation

3.2.1 SDG description and fault propagation analysis of single control loop

Control actions should be considered particularly because they are forced actions that are different from process property itself and they may cause the truncation or misleading of fault propagation. We discuss this problem using the general methods and obtain some special results (Maurya, 2003b, 2006).

In the bottom level, SDG models are established for all kinds of control systems among which the most common and basic one is single PID loop shown as Fig. 4. The deviation e of the set point r and the measurement x_m of the controlled variable x, is inputted into the controller whose output u goes to the actuator and thus effects the controlled plant through the manipulated variable q. Hence they compose a closed loop. Because the controlled variable may be affected by some disturbances or be coupled with other system variables, the exogenous plant and variable x_j are also added. Assume that controlled plant and the controller are both linear amplifiers, i.e. proportion elements, with the positive gain k and k_v respectively. The control law of PID controller is:

$$\begin{cases} u = u_{\mathrm{P}} + u_{\mathrm{I}} + u_{\mathrm{D}} \\ u_{\mathrm{P}} = k_{\mathrm{c}} e \\ (\mathrm{d}/\mathrm{d}t) u_{\mathrm{I}} = k_{\mathrm{c}} e / \tau_{\mathrm{I}} \\ u_{\mathrm{D}} = k_{\mathrm{c}} \tau_{\mathrm{D}} \cdot (\mathrm{d}e/\mathrm{d}t) \end{cases} \tag{15}$$

where, k_P is the positive proportion parameter, τ_{I} and τ_{D} are integral and differential time constant respectively.

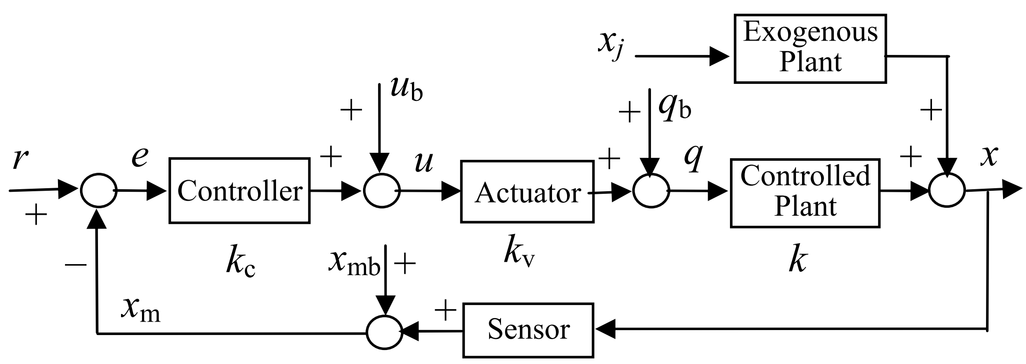

Fig. 4. Block diagram of a feedback control loop

According to the control law, the DAEs of the system are as follows:

$$x_{\mathrm{m}} = x + x_{\mathrm{mb}} \tag{16}$$

$$e = r - x_{\mathrm{m}} \tag{17}$$

$$u_{\mathrm{P}} = k_{\mathrm{c}} e \tag{18}$$

$$(\mathrm{d}/\mathrm{d}t) u_{\mathrm{I}} = k_{\mathrm{c}} e / \tau_{\mathrm{I}} \tag{19}$$

$$u_{\mathrm{D}} = k_{\mathrm{c}} \tau_{\mathrm{D}} \cdot \mathrm{d}e/\mathrm{d}t \tag{20}$$

$$u = u_{\mathrm{P}} + u_{\mathrm{I}} + u_{\mathrm{D}} + u_{\mathrm{b}} \tag{21}$$

$$q = k_{\mathrm{v}} u + q_{\mathrm{b}} \tag{22}$$

$$x = kq + a_j x_j \tag{23}$$

where subscript 'b' denotes bias. There are two perfect matchings between the equations and variables in AE portion, shown as Table 2, whose corresponding SDGs are shown as Fig. 5, in which the nodes with shadow are deviation nodes, arrows with solid and dotted lines denote signs "+" and "-" respectively. It is noted that the node $\mathrm{d}e/\mathrm{d}t$ is an individual node with special function, although it is the derivative of e. In applications, we generally assume that all changes on nodes are step functions, because the SDGs are only used to analyze the qualitative trends. Hence $\mathrm{d}e/\mathrm{d}t$ can be also replaced by e, but its effect is limited in initial response. Here the effect of $\mathrm{d}e/\mathrm{d}t$ on u_{D} is the same as the effect of e on u_{P}, but with shorter duration.

Equations	Matched variables in perfect matching No. 1	Matched variables in perfect matching No. 2
(16)	x_{m}	x
(17)	e	x_{m}
(18)	u_{P}	e
(20)	u_{D}	u_{D}
(21)	u	u_{P}
(22)	q	u
(23)	x	q

Table 2. Perfect matchings between the AEs and variables

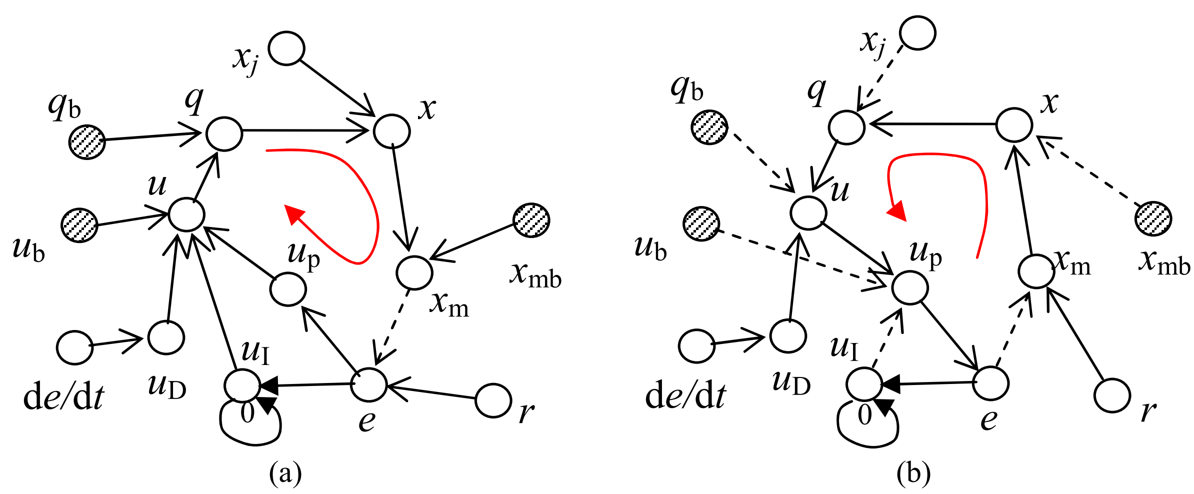

Fig. 5. Two SDGs of the PID control loop. (a) Case 1 (corrected), (b) Case 2 (spurious)

Eq. (23) describes the controlled plant, thus the arc direction should be from q to x according to the physical meaning, which shows the cause-effect relationship, so the case of Fig. 5(b) is removed. Moreover, if the plant shows some dynamic characteristic, for example, the left-hand of the equation is $\mathrm{d}x/\mathrm{d}t$, then the equation becomes a DE, hence there is only one perfect matching, and the case of Fig. 5(b) does not exist any more. Using Fig. 5(a), the initial response can be analyzed, for example, if the set point r increases, e, u_{P}, u, q, x and x_{m} will become "+" immediately, and u_{I} will become "+" gradually because the arc from e to u_{I} is a DE arc. This propagation path $r \rightarrow e \rightarrow u_{\mathrm{P}} \rightarrow u \rightarrow q \rightarrow x \rightarrow x_{\mathrm{m}}$ is consistent with the actual information transfer relations. Thus when we only consider the initial response of the system, the SDG of this control loop is obtained by transforming the blocks and links in block diagram into nodes and arcs while keeping the direction. However, in this example, no matter whether the case of Fig. 5(b) is reasonable, the analysis results of initial response by the two SDGs are the same because there are no positive cycles within them. We summarize the following rule:

Rule 1: The fault propagation path of the initial response in a control loop is the longest acyclic path starting from the fault origin in the path "set point → error → manipulated variable → controlled variable → measurement value → error", which is consistent with the information flow in the block diagram.
Final response is easier. The left-hand side of Eq. (19) is zero, so $e = 0$ in the steady state, which can be obtained from the concept. Hence u_P and u_D are both zeros. The above DAEs can be transformed into:

$$x_m = x + x_{mb} \tag{24}$$

$$x_m = r \tag{25}$$

$$u = u_I + u_b \tag{26}$$

$$q = k_v u + q_b \tag{27}$$

$$x = kq + a_j x_j \tag{28}$$

Now the perfect matching is exclusive and the corresponding SDG is shown as Fig. 6 that is the simplification of Fig. 5(b). There are two fault propagation paths: $r \to x_m \to x \to q$ and $x_j \to q \to u \to u_I$. If the set point r increases, then x_m, x, q, u and u_I will all increase in the steady state as long as the control action is effective. However, if only x_{mb} increases, then x_m will not be affected, but x will increase, that is the action of the control loop. We find that the Fig. 5(b) also makes sense for it reflects the information transfer relation in steady state. From the viewpoint of physical meaning, when control loop operates, the controlled variable is determined by the set point, and the controller looks like an amplifier with infinite gain, whose input equals to zero and whose output is determined by the demands. This logical transfer relation is opposite to the actual information relation.

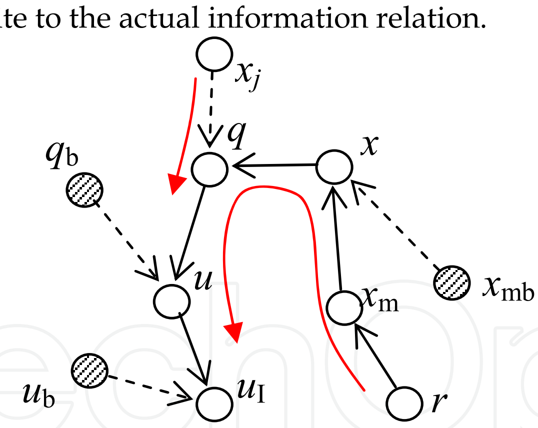

Fig. 6. Steady-state SDG of a PID control loop

Because the D action is only effective in the initial period, the fault propagation path of PI control is the same as the above one. Because of I action, some variables show compensatory response, for example, the response of x_m due to x_{mb} is limited in the initial stage. If there is only P action, then e is not zero in the steady state, thus u_I and related arcs in Fig. 5(a) are deleted, and both the initial response and steady-state response can be analyzed with this graph.
The rule of fault propagation analysis in steady state can be summarized as follows:

Rule 2: The fault propagation path of the steady-state response in a control loop is the path "set point → measurement value → controlled variable → manipulated variable" and "exogenous variable → manipulated variable".
When control loop operates, the above analysis shows the fault propagation principles due to the output deviation of sensor, controller, actuator and other exogenous variables. When control loop does not operate, there are two cases: (1) structural faults, e.g. the failure of sensor, controller or actuator causes the break of some arcs and the control loop becomes open, (2) excessive deviation causes the controller saturation, leading to the I action cannot eliminate the residual and let $e = 0$, which is similar with the P action case.

3.2.2 SDG description and fault propagation analysis of various control systems

Based on the above analysis of PID control loop, other control loops can be modeled as SDGs by the extension, combination, or transformation of the above SDG. Fault nodes are added according to the actual demands. Based on these models, fault propagation can also be shown explicitly.
Feedforward control is a supplement of feedback control. It is very familiar in actual cases, but it is easy to be treated according to the foregoing methods because it composes paths but not cycles, not leading to multiple perfect matchings.
Split-range control means the different control strategies are adopted in different value intervals. Here the sign of the arcs or even the graph structure may change with the variable values, which is realized by several controllers in parallel connection. This case is very hard for SDG to deal with. We have to do some judgments as making inference, and modify the structure or use conditional arcs to cover all the cases (Shiozaki et al., 1989).
Cascade control can be regarded as the extension of single loop case. It can be solved directly by AEs, or by the combination of two single loops. For example, the cascade control system in Fig. 7 has the steady-state SDG as shown in Fig. 8, where the controlled variable of the outer loop u_1 is the set point of the inner loop r_2.

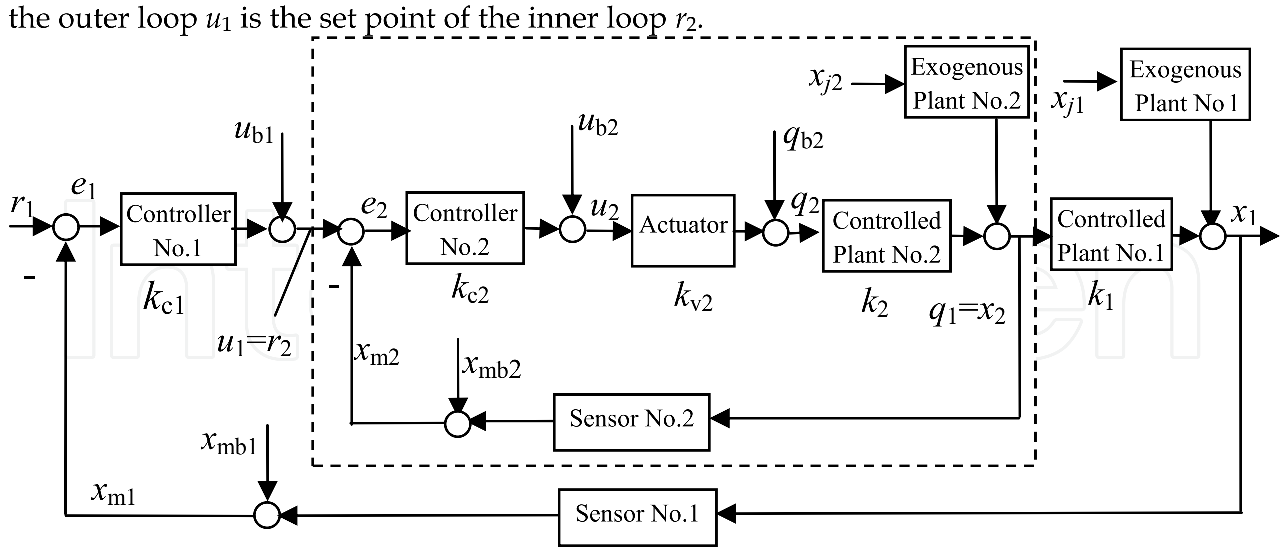

Fig. 7. Block diagram of a cascade control system

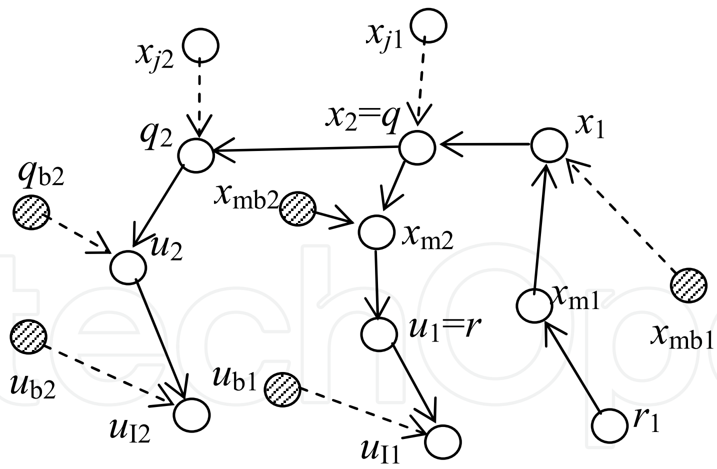

Fig. 8. Steady-state SDG of a cascade control system

Similar control methods are ratio control, averaging control, etc. Fig. 9 is a dual-element averaging control system whose objective is to balance two variables – level and flow, the block diagram of which is shown as Fig. 10. $P_x=P_L-P_F+P_S+c$, where P_x is the pressure signal of the adder output, P_L is the level measurement signal, P_F is the flow measurement signal, P_S is a tunable signal of the adder. In the simplest case, flow process and its measurement are both positive linear elements, and the level process is a negative linear element, so the steady-state SDG is shown as Fig. 11. Although there are several perfect matchings, SDG has only a negative cycle, thus we can analyze the fault propagation principle through the directed paths.

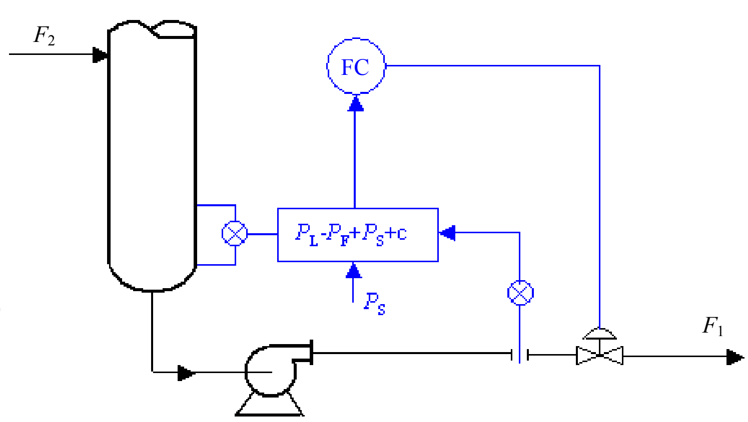

Fig. 9. A dual-element averaging control system

Thus we conclude:

Rule 3: The fault propagation path in a control system in steady state can be combined from the ones of single-loop by combining the same nodes and adding arcs by transforming AEs. In an industrial system, control systems play a special but important role. They compose information flow cycles in initial response but result in different flow in steady state response. Fig. 12 shows a system with a control loop. According to the above rules, the bias in x_1 propagates along the forward path (blue) in initial response while against the feedback path in steady state response.

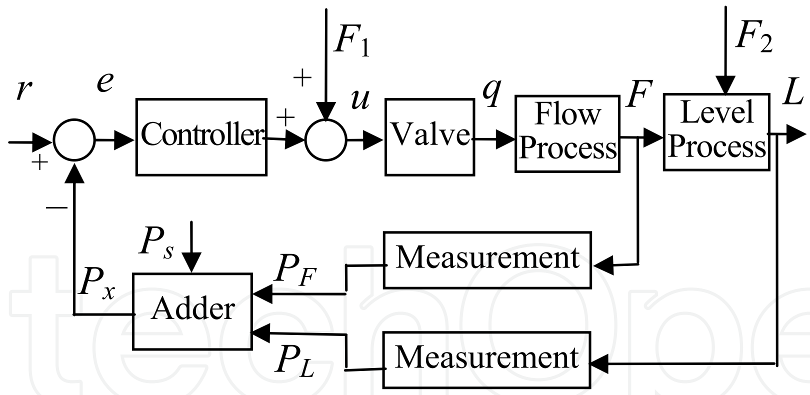

Fig. 10. Block diagram of a dual-element averaging control system

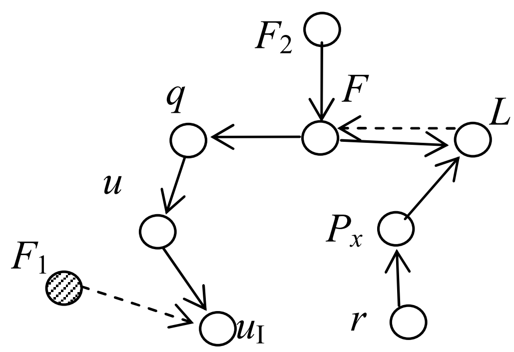

Fig. 11. Steady-state SDG of a dual-element averaging control system

The bias in x2 propagates along two paths until x3 and x4 in initial response, while PV and x4 restore to normal in the steady state because the steady state SDG changes the structure and directionality of the graph and thus PV becomes a compensatory variable.

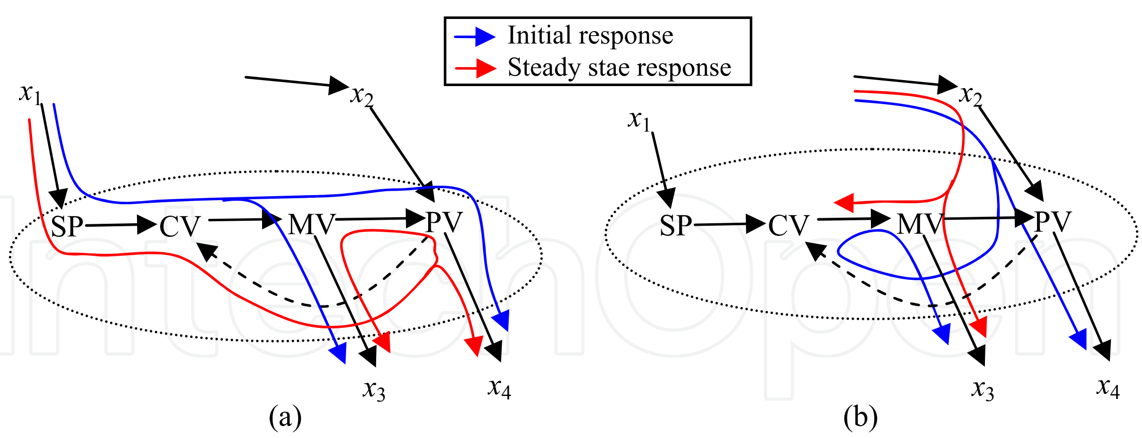

Fig. 13. Control system's effect on fault propagation in a system. (a) Bias in x_1, (b) Bias in x_2

3.2.3 Example

In a boiler system, the three-element control of the boiler water level is widely used, in which the main controlled variable is water level. If we take steam flow and inlet flow into account, the control system is a feedforward-cascade system, as shown in Fig. 13. In the initial stage of the disturbance, the SDG is shown as Fig. 14(a), which can be derived by original DAEs. Certainly the initial fault influence follows this SDG. The control action, however, is enrolled and some deviations are restored to the normal region after a complex intermediate process. If the control action is successful, the fault may be blocked in the control loop and does not spread any more. But for some kinds of faults, the situation is different, because the control action makes the fault propagation path change. According to the method in the foregoing sections, we can construct the backbone (ignoring the bias nodes) of the steady-state SDG model as Fig. 14(b) which is quite different from Fig. 14(a). Similar with Fig. 9, other fault nodes can be added to Fig. 14(b) and thus we can find the steady-state fault propagation paths.

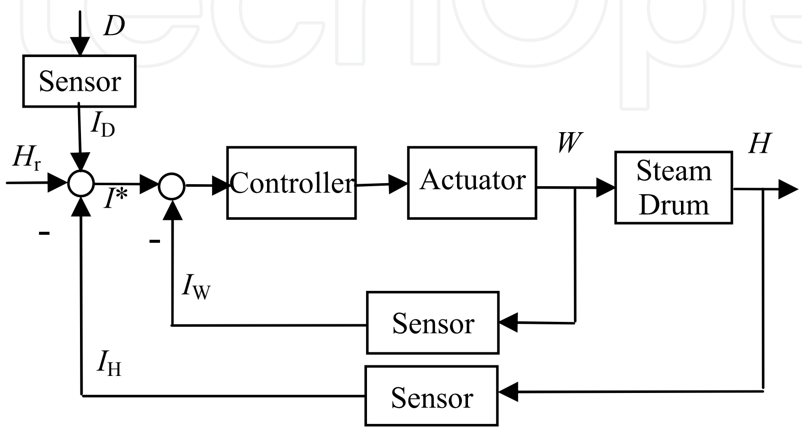

Fig. 13. Three-element control system

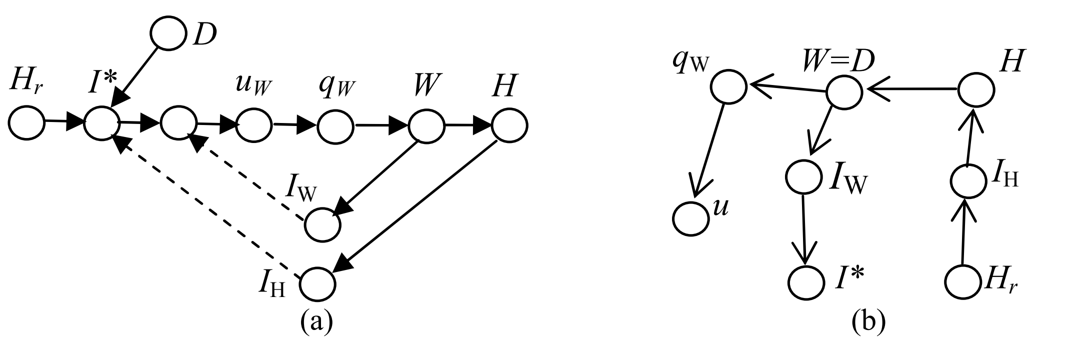

Fig. 14. SDGs of the three-element control system. (a) Initial response, (b) Steady state

3.3 Inference Approaches

In safety area, fault diagnosis and hazard assessment, especially hazard and operability analysis (HAZOP) are two different tasks. The former is to correctly find and identify the fault origin that is the cause leading to the symptom when fault occurs. It is based on measurements and is real-timed. While the latter, hazard assessment is to an off-line analysis whose purpose is to find the possible hazards due to all various causes. For this reason, we assume a series of departure nodes as fault origins, then analyze the possible consequences that are all the triggered departure nodes. Both fault diagnosis and hazard assessment need the interior mechanism of the system to express how the faults propagate. Thus the SDG model can be employed.

3.3.1 Graph traversal approach

The most common algorithm for searching the fault origin is depth-first traversal on the graph (Iri et al., 1979), which is a kind of efficient fault inference for both the single and multiple fault origin cases (Zhang et al., 2005). Its theoretical basis is nodal balance in Eq. (14). A depth-first traversal algorithm constructs a path by moving each time to an adjacent node until no further arcs can be found that have not yet been visited, the implementation of which is a recursive procedure.

For HAZOP purpose, forward traversal is applied from the assumed origin to predict all the variables based on consistency, which is deductive reasoning (Venkatasubramanian et al., 2000; Yang & Xiao, 2006b). For the fault detection purpose, backward traversal is applied within the causal-effect graph to find the maximal strongly connected component (Iri et al., 1979), which is abductive reasoning. Actually, the whole procedure includes two steps:

Step 1: Trace the possible fault origins back along the arcs.

Step 2: Make forward inference from these nodes to screen the candidates to choose which one is the real or most probable fault origin.

The time complexity of a traversal search is $O(n^2)$ in which n denotes the node number in the graph. When the system scale increases, the time for a traversal is too long to meet the demands of fault detection. Thus the model structure should be transformed from a single-level one to a hierarchical one. By this way, the search is first performed in the higher level to restrict the fault origin in a sub-system. Then the search is performed in the sub-graph of this sub-system.

For the hierarchical model, hierarchical inference from top to bottom is obtained naturally. The graph traversal is performed firstly in the higher level finding the possible super-node that includes the fault origin. Next perform the graph traversal in the lower level to restrict the possible location of the root cause. Assume the sub-system contains m control systems, and each control system contains k variables, then the time complexity of a traversal in a single-level model is $O(m^2k^2)$, and the time complexity in a 2-level model is $O(m^2+k^2) \ll O(m^2k^2)$. Thus the fault analysis in a hierarchical model has much higher efficiency.

Here the number of fault origin is assumed to be only one, that is, the reason that leads to the fault is only one (Iri et al., 1979). This is reasonable because multiple faults seldom appear at the same time (Shiozaki et al., 1985). For multiple fault origin cases, minimal cut sets diagnosis algorithm was presented (Vedam & Venkatasubramanian, 1997), where all possible combinations of overall bottom events should be input into the computer to explore and those which make the top events appear are the cut sets. This algorithm has the distinct disadvantage of low efficiency because of exponential explosion.

3.3.2 Other improved approaches

In order to utilize the system information more sufficiently, Han et al. (1994) used fuzzy set to improve the existing models and methods, but their method is not so convenient for on-line inference and is not applicable for dynamical systems. Some scholars introduced temporal evolution information such as transfer-delay (Takeda et al., 1995; Yang & Xiao, 2006a) and other kind of information into SDG for dynamic description. Probability is also proposed to model the system, which uses conditional probabilities of fault events to describe causes and effects among variables (Yang & Xiao, 2006c). Hence the inference is respect to the fault probability.

We can use Bayesian inference on the graph to calculate the probabilities; it is a direct method. Suppose that the node set of the probabilistic SDG is $V = E \cup F \cup H$, in which E is the subset of evidence nodes whose value or probabilities are known, F is the subset of query nodes whose probabilities are to be computed, and H is the subset of hidden nodes which is not cared about in the inference. The inference process of is to compute the conditional probability of x_F given the known x_E.

$$p\left(x_F \mid x_E\right) = \frac{p\left(x_E, x_F\right)}{p\left(x_E\right)} \tag{29}$$

where

$$p\left(x_E, x_F\right) = \sum_{x_H} p\left(x_E, x_F, x_H\right) \tag{30}$$

$$p\left(x_E\right) = \sum_{x_F} p\left(x_E, x_F\right) \tag{31}$$

To solve this problem, Bayesian formula and its chain rule should be used adequately, and also the junction tree algorithm can be used for multiple fault origin cases. This method could be used where there are distinct random phenomenon, both for fault detection and HAZOP (Yang & Xiao, 2006b), but the cycles in SDGs should be handled. The algorithm is the combination of depth-first search and junction tree algorithm, written as pseudo code:

```
BEGIN
 INTEGER i
 PROCEDURE DFS(v,u); COMMENT v is the father of u
 BEGIN
   NUMBER(v):=i:=i+1
   Calculate the probability of the father of v ; COMMENT junction tree algorithm
   FOR v has a father w with the probability more than the threshold DO
     BEGIN
      IF w is not yet numbered THEN
        BEGIN
        DFS(w,v);
     END;
  END;
  i:=0;
  DFS(s,0)                    ; COMMENT s is an abnormal variable node
END;
```

On the other hand, rule-based inference (Kramer & Palowitch, 1987) is applicable when expert system is available. This method can be used to improve the inference accuracy with the appropriate rule description and operation. Rough set theory provides an idea of handling vague information and can be used to data reduction, thus it can be introduced to the fault isolation problem (a kind of decision problems) to optimize the decision rules. The decision algorithm is proposed by Yang & Xiao (2008a), in which the generation and reduction method of the rules are related to the structure of the SDG model. The main steps are listed as follows:

(1) List all the possible rules as *Table A* (as Table 3), with each row denoting a rule $\varphi \to \psi$, where φ denotes the values of the condition attributes are assumed and ψ denotes the decision to be obtained. For convenience, we can give each attribute value a notion.

Attributes **Q** / Objects **X**	Condition attributes **C**	Decision attributes **D**

Table 3. The framework of a decision table

(2) Try to delete each condition attribute in turn and test the consistency of the formula and obtain the reducts and the core. Delete all the elements except the cores and get *Table B*. There are several methods to test the consistency. For example,
  (a) Each condition class $E \in X \mid IND(\mathbf{C})$ has the same decision value.
  (b) For each object x, the condition class covering x is contained in the decision class covering x.
  (c) For every two decision rules $\varphi \to \psi$ and $\varphi' \to \psi'$, we have $\varphi = \varphi' \to \psi = \psi'$.

(3) Calculate the reducts of each rule by use of Table B, and get *Table C*.

(4) Delete redundant rules and thus get *Table D*.

(5) Educe the rules and the decision algorithm according to Table D.

The authors combine the algebraic and logical expression ways to achieve the purpose. Moreover, due to the convenience of expressing granularity, the decision algorithm is still applicable when the types of the faults of concern are changed or reformed.

4. Sensor Location Problem Based on SDGs

4.1 Performance Criteria of Fault Detection

4.1.1 The dynamic SDG and fault reachability

In actual systems, the fault propagation needs time, which effects the fault detection performance. So we take into account the fault propagation time for each branch and form the dynamic SDG (Yang & Xiao, 2008b). If the variable denoted by the node n_1 has a direct influence on the variable denoted by n_2, and after a time period for the fault propagation the fault revealed, then we define this time period τ (n_1, n_2) as the fault propagation time between n_1 and n_2, as shown in Fig. 15. Obviously, we have τ (n_1, n_2) ≥0. A path starting from n_1 and ending at n_m (denoted as l ($n_1 \mapsto n_m$)) holds the overall fault propagation time τ ($n_1 \mapsto n_m$) which is the summation of time τ (n_i, n_j) of each branch in this path, as shown in Fig. 15. Note that this is a simplified treatment, which fits the case of pure propagation delay, but when the dynamic properties are complex, the overall time may slightly decrease due to the effects of intermediate transients (Yang & Xiao, 2006a).

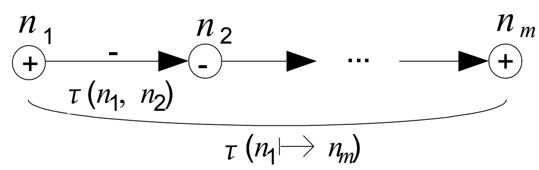

Fig. 15. Propagation time of a consistent branch and a consistent path

Because the nodes in SDGs are classified into two types – variables and fault origins, we denote them as n_is and f_js respectively. When a fault occurs, it is propagated along the consistent paths together with the time progress.

Definition 4: Starting from the fault node f, after the time t, the set of nodes affected by f is

$$R(f,t) = \{m : \exists l(f \mapsto m) \text{ and } \tau(f \mapsto m) \leq t\} \tag{32}$$

where t is the fault propagation time. If $n \in R(f, t)$, then we say, node n is *reachable* from fault f in time period t. Obviously, when time proceeds, the set of affected nodes expands, thus $R(f, t_1) \subseteq R(f, t_2)$, if $t_1 < t_2$.

The basic criteria of fault detection are detectability and identifiability to assure the faults be detected and identified from each other. The concepts here are the extension of the concepts in the framework of the SDG.

4.1.2 Fault detectability and detection time

A fault should be detected by at least one sensor in a short enough time period. Below is the definition.

Definition 5: If there exist at least one sensor located in the nodes of $R(f, t)$ (measuring the corresponding variables), then we say that the fault f is *detectable* in the time period t. The time needed to detect a fault by these sensors is called the detection time $T_D(f)$.

For each sensor, the time needed to detect a fault f can be calculated by shortest path algorithm. Among all these sensors, the shortest time is recorded as $T_D(f)$. The number of nodes with sensors in $R(f, t)$ is called the degree of detectability.

Based on the traditional SDG, only leaf nodes are needed to consider whether or not to locate sensors (Raghuraj et al., 1999). Then we have the following theorem.

Theorem 1: Based on the SDG, disregarding the cases that some variables cannot be measured, sensors need to be located only on the leaf nodes.

Corollary 1: In the framework of dynamic SDG with propagation time, the sensors need to be located on only leaf nodes of $R(f, t)$.

4.1.3 Fault identifiability and identification time

Different faults have different behaviors. Represented in the SDG, the reachable nodes are different. So we must put sensors on these different nodes to identify the different faults. Below is the definition.

Definition 6: If there exist at least one sensor on the nodes of $R(f_1, t)$ (measuring corresponding variables), and these sensor nodes are not within the nodes of $R(f_1, t)$, in other words, if there are sensors in the nodes of $I(f_1, f_2, t) = R(f_1, t) \cup R(f_2, t) - R(f_1, t) \cap R(f_2, t)$, then we say that the faults f_1 and f_2 are *identifiable* in the time period t. The time needed to identify two faults by these sensors is called the identification time $T_I(f_1, f_2)$.

Detectability and identifiability are two independent concepts. We can understand easily, when two faults are both detectable, they may not be identifiable. On the other hand, identifiability does not imply detectability generally, because we can place only one sensor to identify them too. But usually we assume that only when the faults are detectable, they can be considered for identifiability. Thus the identifiability condition is stronger. In Definition 3, I (f_1, f_2, t), for two identifiable faults, must have more than one element. The number of nodes with sensors in I (f_1, f_2, t) is called the degree of identifiability. Besides, we have

Proposition 1: $T_I(f_1, f_2) \geq \max\{T_D(f_1), T_D(f_2)\}$.

Proposition 2: The number of elements in I (f_1, f_2, t) is not necessarily increasing monotonically with time t.

It should be noted that the signs of the nodes and branches can help identify different faults because some sensors are not only able to activate the alarm, but also indicate the direction of the departure from the normal values. For this case, we could change a node into two, one shows the higher reading, another shows the lower reading (Wilcox & Himmelblau, 1994). Then the above definition and the following rules can be applied.

4.1.4 Detectability and identifiability with multiple faults

Sometimes we also need to deal with the case of multiple simultaneous faults. It can be dealt with by node set transformation.

Here we take two faults as an example. If the faults f_i and f_j occur at the same time, their reachable node set is $R_i \cup R_j$, so we can take these C_n^2 node sets to be considered besides the sets of R_i, then the problem is transformed into the detectability and identifiability problems with a single fault.

Obviously, if each fault can be identified, but when several faults occur at the same time, they are not assured to be identified. How about the inverse proposition?

Theorem 2: If the case of n simultaneous faults can be identified, then the case of less than n simultaneous faults can be also identified.

4.1.5 Fault detection reliability

Detectability and identifiability are necessary conditions for fault detection. However the sensor readings are not always reliable, which affects the reliability of fault detection. Let F_is (i=1, 2, …, n) and S_js (j=1,2,…,m) denote system faults and process variables measured by sensors individually. They can be shown as a bipartite graph with all the arcs directed from the fault set to the process variable set as shown in Fig. 16. Based on the detectablity criterion, there should be at least one arc departing from every fault node, and based on the identifiability criterion, the connected sensor nodes of different fault nodes should be different. The fault occurrence probabilities of the fault F_i is f_i, while the sensor missed alarm rate and false alarm rate of variable S_j is u_j and v_j. The influence relation from fault F_i to sensor S_j is denoted by reachability d_{ij} (0 or 1) where 1 means reachable and 0 means unreachable. Because of the causal relations between process variables, the reachability includes direct and indirect influences.

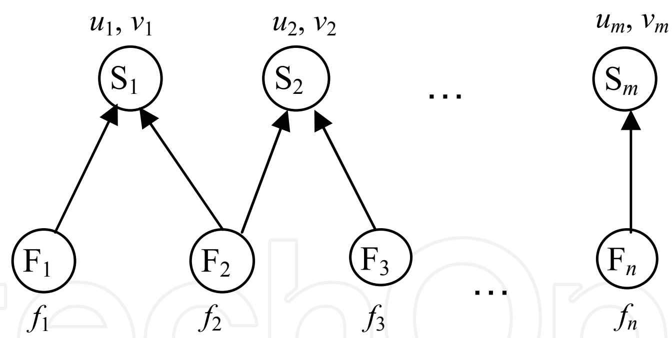

Fig. 16. Bipartite graph to show the relations between faults and sensors

As shown in Fig. 17, the confusion matrix reflects the true/false classification of alarms (Izadi eta al. 2009). The entries in the matrix are the number of true alarms (TA), false alarms (FA), missed alarms (MA) and true no-alarms (TN). These numbers can be obtained by experiments. The missed alarm rate of sensor S_j is u_j which can be calculated by MA/(TA+MA), and false alarm rate of v_j can be calculated by FA/(FA+TN). These rates are determined by the sensor quality and the threshold selection.

Hypothesized Class	True Class: Fault	True Class: No fault
Alarm	True Alarms (TA)	False Alarms (FA)
No alarm	Missed Alarms (MA)	True No-alarms (TN)

Fig. 17. Confusion matrix to show the terminology of missed alarms and false alarms

For each fault F_i, we should minimize its probability of not being detected. Because it is propagated to many other variables on which the sensors can also detect it, the undetectability of F_i occurs only when all the variables miss alarms. Besides, the redundant sensors on the same variables are also helpful for the improvement. We define the undetectability probability (Bhushan & Rengaswamy, 2002) of F_i as

$$U_i = f_i\left(\prod_{j=1}^{m}\left(u_j\right)^{d_{ij}x_j}\right) \tag{33}$$

where x_j is the integer number of sensors put on the variable j. If there is no sensor on variable j, x_j is zero. Obviously, when x_j with the corresponding nonzero d_{ij} increases, U_i decreases. So adding sensors will increase the reliability.
On the other hand, we think about the false alarm problem. For the variable S_j, adding a sensor with false alarm rate v_j (with respect to fault F_i) will be accompanied with the increase of the following false alarm probability

$$V_j = v_j \left(\prod_{i=1}^{n} (1 - f_i)^{d_{ij}} \right) \quad (34)$$

which means the sensor reading gives the alarm even though no faults occur.
The calculations of missed alarms and false alarms are dual problems that adding sensors will reduce the undetectability whilst increasing the false alarm probability. Here the false alarm probability reflects the influence of a sensor's false alarm on the whole system.

4.2 Sensor Location Based on Fault Detectability and Identifiability

The purpose of the sensor location problem is to choose sensors and design the sensor location to meet the demands of fault detection. Neglecting the reliability problem, here we deal with the detectability and identifiability problems. In the framework of the static SDG, the problem can be solved directly. But in the framework of dynamic SDG, the arising times of various faults are various, and different faults may interact, so it is hard to analyze all the cases of fault propagation or even solve the sensor location problem in advance. One possible way is to embed the sensor location problem in the forward inference process as the following algorithm:

(1) Add fault node f_i to the evidence node set E and the reachable node set R_i. Set the inference system time T_{sys} to zero.
(2) Check if the evidence node set is empty. If it is empty, then go to the end, otherwise go on.
(3) From the evidence nodes, choose the nodes in the reachable node set for one forward step, and add them to the reachable node set R_E of the evidence nodes. Meanwhile, update their detection time $T_D(f_i)$ (detection time of the starting node of the branch plus the propagation time on the branch).
(4) From the nodes in R_E, choose one with the shortest detection time T_k and the nodes to be updated, N_{T_k}, at time T_k.
(5) $T_{\text{sys}} = T_{\text{sys}} + T_k$. Make forward inference from all the nodes in N_{T_k} for one step.
(6) Add N_{T_k} to the evidence node set E and reachable node set R_i.
(7) If an evidence node whose one-step reachable nodes are all updated, then delete this node from E.
(8) Placing sensors in the reachable node set R_i can assure the detectability of fault i, and placing sensors in $R_i \cup R_j$ -$R_i \cap R_j$ can assure the identifiability of fault i and j.
(9) If a new fault has occurred, then add its corresponding node to the evidence node set E, and set T_{sys} as the current time. Go to step (2).

Note that the treatment in step (3) is not accurate because the detection time is just approximate. So we often increase the threshold of the degree of detectability and identifiability to assure performance is optimal.

4.3 Sensor Location Based on Fault Detection Reliability

The two criteria, detectability and identifiability should be met at first when deciding the sensor location. Besides above algorithm, Yang and Xiao (2008b) also proposed some useful rules to solve this problem in consideration of the propagation time, which is a stricter requirement than that mentioned above. The sensor location obtained has the minimum number of sensors required for fault detection. Since the increase of sensors will not destroy these criteria, the following optimization algorithm should be based on this location and try to find the crucial variables for putting additional sensors.

In the trade-off between false alarms and missed alarms, missed alarms are often considered to be more important because we do not want to lose a real fault. Thus the algorithm handles this criterion first. Meanwhile, we hope the false alarm rates to be as small as possible, so we integrate the treatment of false alarms into the whole algorithm.

If we consider all the faults, then we want to minimize the total undetectability probabilities for all the faults, each one of which is a probability that no sensors give the alarm for the corresponding fault. Thus we have the following optimization problem:

$$\min_{x_j}\left[\sum_{i=1}^{n} U_i\right] \tag{35}$$

This optimization problem cannot be solved at once (by branch and bound method or other methods) for the following reasons. First, this problem does not have a continuous solution space; instead it is an integer programming problem. Thus we should update the solution (x_j, j=1,…,m) once at an integer. Secondly, the problem has constraints. For example, putting a sensor on a variable needs some cost, and the total cost should be limited within a range, so we have

$$\sum_{j=1}^{m} c_j x_j \le C_0 \tag{36}$$

where c_j is the cost to be paid when putting a sensor on variable j, and C_0 is the cost limit. Thirdly, the initial value of the problem is obtained according to the criteria of detectability and identifiability, and the x_js should not be negative, which can be regarded as another constraint. Sometimes we have more constraints such as the number limit of sensors. This algorithm is just used to reduce the undetectability by adding sensors at critical location. Thus the problem is solved by an iterative algorithm, and within each step we should only add 1 to one of the x_js and check the constraints. This is a heuristic algorithm.

Besides, false alarm problem can also be formalized as an integer optimization problem:

$$\min_{x_j}\left(\sum_{j=1}^{m} x_j V_j\right) \tag{37}$$

but this problem is accompanied with the undetectability optimization problem and is less important for most cases. Thus we do not take it as an individual problem but as a supplement to the above problem expressed by the following formulation

$$\min_{x_j}\left[\sum_{i=1}^{n} U_i + \alpha \sum_{j=1}^{m} x_j V_j\right] \tag{38}$$

where α is a constant coefficient.

When trying to reduce the undetectability by adding a sensor, one is concerned not with the total number of missed alarms but the number for each fault. Thus the summation in Eq. (35) can be replaced by a weighted summation, where the weights correspond to the importance. The weights are not impersonal or rational to obtain, so we can alternatively use the maximization to deal with the bottleneck which is the fault with maximal undetectability. Hence we have the following optimization problem as a combination of a minimaxization and a linear minimization

$$\min_{x_j}\left[\max_i\left(U_i\right)+\alpha\sum_{i=1}^{m}x_jV_j\right] \tag{39}$$

subject to

$$\sum_{j=1}^{m}c_jx_j\le C_0,\ \ x_j\in Z^+\cup\{0\} \tag{40}$$

If less attention is paid to the false alarm rate, we can take its optimization as a constraint and just set a limit V_0 instead of optimization. Then we get the simplified algorithm:

(1) Initialization:
   (a) Get f_i, u_j and v_j by a priori knowledge and measurements.
   (b) Get d_{ij} from SDG or reachability matrix.
   (c) Get the minimal x_js according to the criteria of detectability and identifiability as the starting point.
   (d) Calculate V_j by Eq. (34).
   (e) Calculate V by summation of all the V_js with x_j is not 0.
(2) Calculate U_i and select the maximal one U_I.
(3) Let the set of js with d_{Ij} is 1 as A_I = max{j | d_{Ij}=1}.
(4) Select the maximal u_J from A_I, i.e. u_J = max A_I. If A_I is empty, stop. If there is more than one maximum element, select the one with smallest V_j.
(5) Put a sensor on variable J, $x_J \leftarrow x_J + 1$.
(6) Update the false alarm rate $V \leftarrow V+V_j$, and see if it is tolerable. If so, go on; if not, delete J from A_I and go to step (4).
(7) Check the constraints. If they are met, go on; if not, delete j from A_I and go to step (4).
(8) Go to step 2 and update the undetectability.

The algorithm is illustrated as a flow chart in Fig. 18.

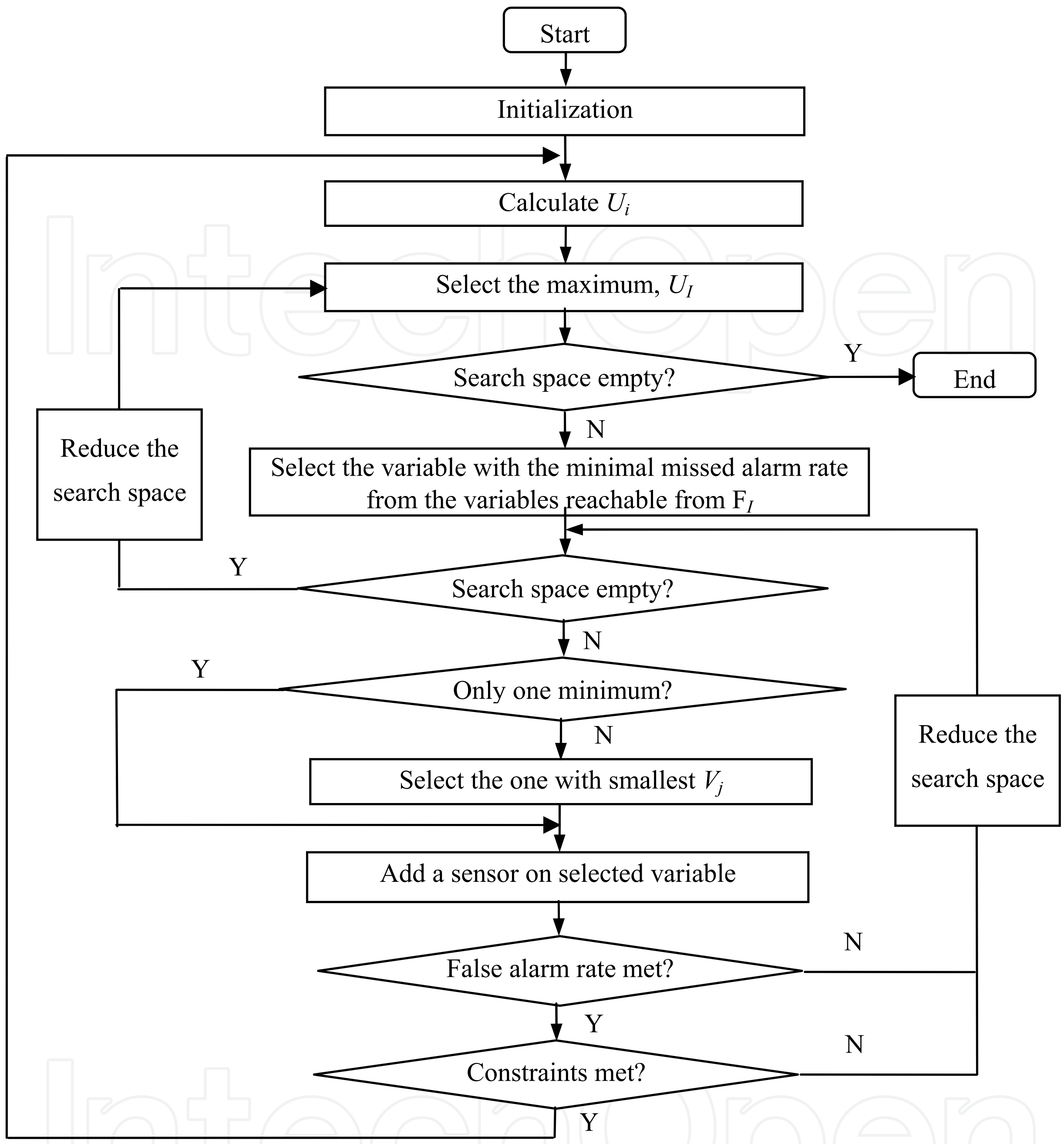

Fig. 18. Flow chart of the optimization algorithm.

5. Case Study

We take a 100 MW generator set process in a power plant as an example, which is composed of a typical natural-circulation steam boiler and a turbine. The system is operated and controlled by a DCS of MACS-II.

The process can be divided into several sub-systems such as water & steam system, coal & air system, and turbine system. The core flowsheet is shown in Fig. 19.

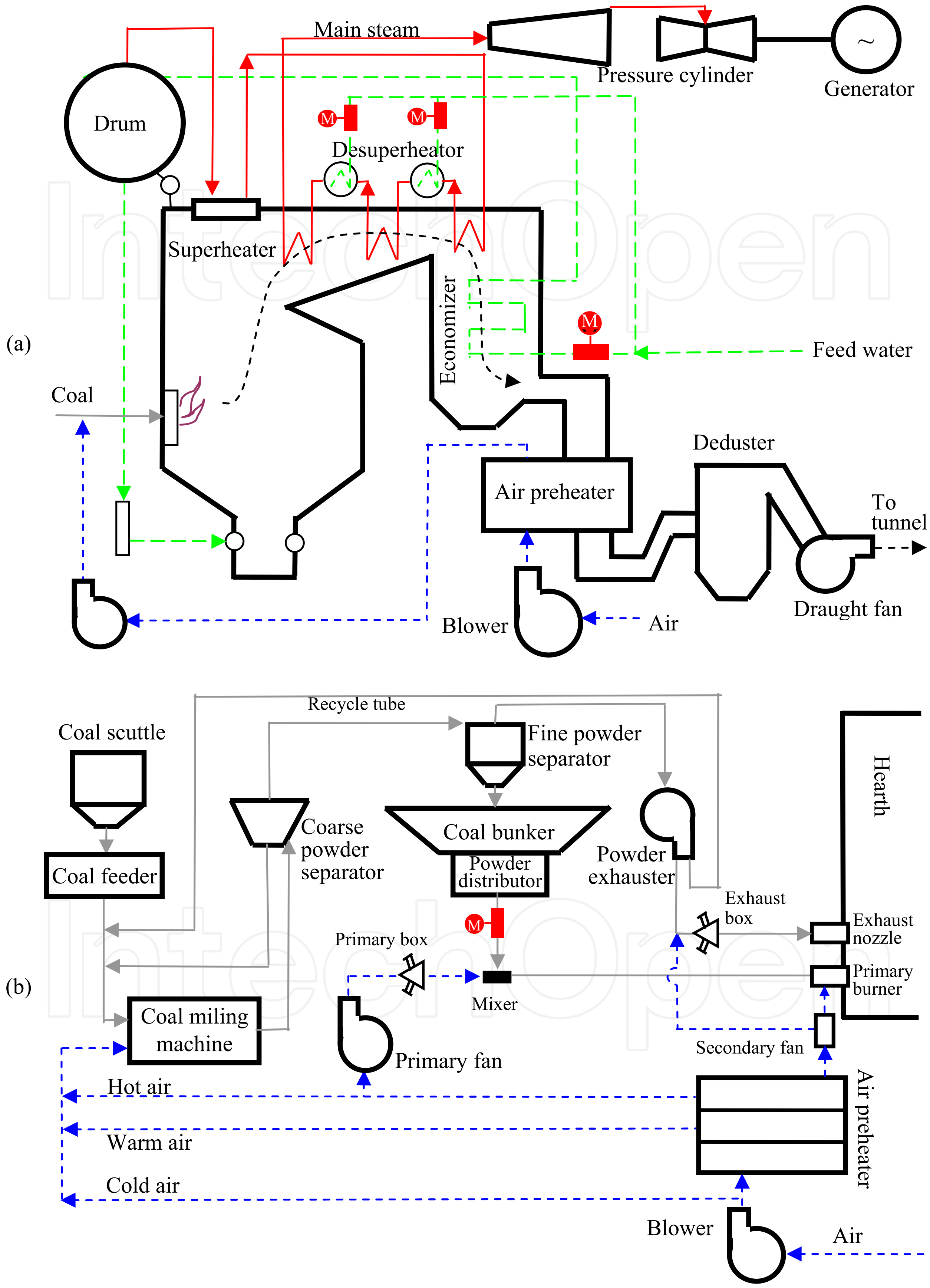

Fig. 19. Generator set process schematic. (a) Boiler and turbine, (b) Coal and air

5.1 SDG Model Description of the Generator Set Process

As listed in Table 4, the major variables are controlled in separated systems. Note that the control system of oxygen percentage in the smoke is not included here because it is so complex that it is usually operated manually. And the control systems of water level of condensator, deoxidizor and exchangers are also not included because they are operated in independent single control loops that can be separated from the whole graph. Thus, the top-level SDG model is composed of several super-nodes in which we are only concerned about the major ones corresponding to the 5 controlled variables.

Control system	Description	Controlled variable	Tag name	Operating value
S1	Water level control by manipulating inlet water flow	Water level of the top steam drum	L1	0 mm
S2	Overheated steam flow control by throttle valve to maintain the turbine rev	Turbine torque	M	
S3	Overheated steam temperature control by manipulating the cooling water valve	Overheated steam temperature	T1	535 ^{o}C
S4	Overheated steam pressure control by manipulating the transducer frequency of coal powder distributor	Overheated steam pressure	P2	8.83 MPa
S5	Hearth pressure control by manipulating the fan baffle	Hearth pressure	P0	-50 Pa

Table 4. Controlled variables in the generator set process

The other variables in the process are listed as Table 5. We model the middle-level SDG of the system shown as Fig. 20. In the bottom level, control systems in S2–S5 are single loops in nature, whose SDGs are shown as Fig. 21(a)-(d).

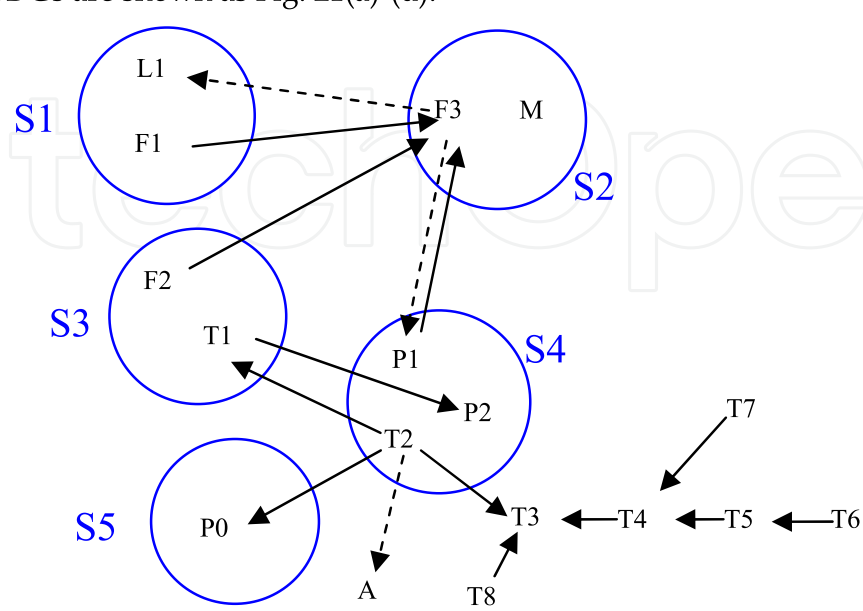

Fig. 20. Middle-level SDG model of the generator set process

Tag name	Variable	Operating value
F1	Inlet water flow	360 t/h
F2	Cooling water flow	10 t/h
F3	Overheated steam flow	370 t/h
F4	Exit smoke flow	
P0	Furnace negative pressure	
P1	Steam drum pressure	10 Mpa
P2	Superheated steam pressure	
P3	Inlet water pressure	12 Mpa
P4	Cooling water pressure	12 Mpa
P5	Primary total air pressure	1.4 kPa
P6	Blower exit air pressure	
P7	Coal powder exhauster exit air pressure	
P8	Coal powder exhauster inlet air pressure	
T2	Hearth temperature	
T3	Exit smoke temperature	150 ^{o}C
T4	Primary air temperature	
T5	Primary air exit temperature	
T6	Primary air inlet temperature	
T7	Coal milling machine exit air temperature	68 ^{o}C
T8	Inlet water temperature	215 ^{o}C
V1	Transducer frequency of coal powder distributor	
V2	Turbine rev	
N1	Turbine power	100 MW
N2	Turbine load	100 MW
A	Oxygen percentage in the smoke	5.8 %
C1	Cooling water valve	
C2	Primary fan baffle	
C3	Blower baffle	
C4	Coal powder exhauster baffle	
C5	Draught fan baffle	
C6	Inlet water valve	
C7	Main throttle valve	

Table 5. Other variables in the generator set process

In S1, the three-element control of the water level is used, in which the main controlled variable is L1. If we take steam flow F3 and inlet water flow F1 into account, the control system is a feedforward-cascade system, as shown in Fig. 22(a). In the initial stage of the disturbance, the SDG is shown as Fig. 22(b), which can be derived by DAEs. Certainly the initial fault influence follows this SDG.

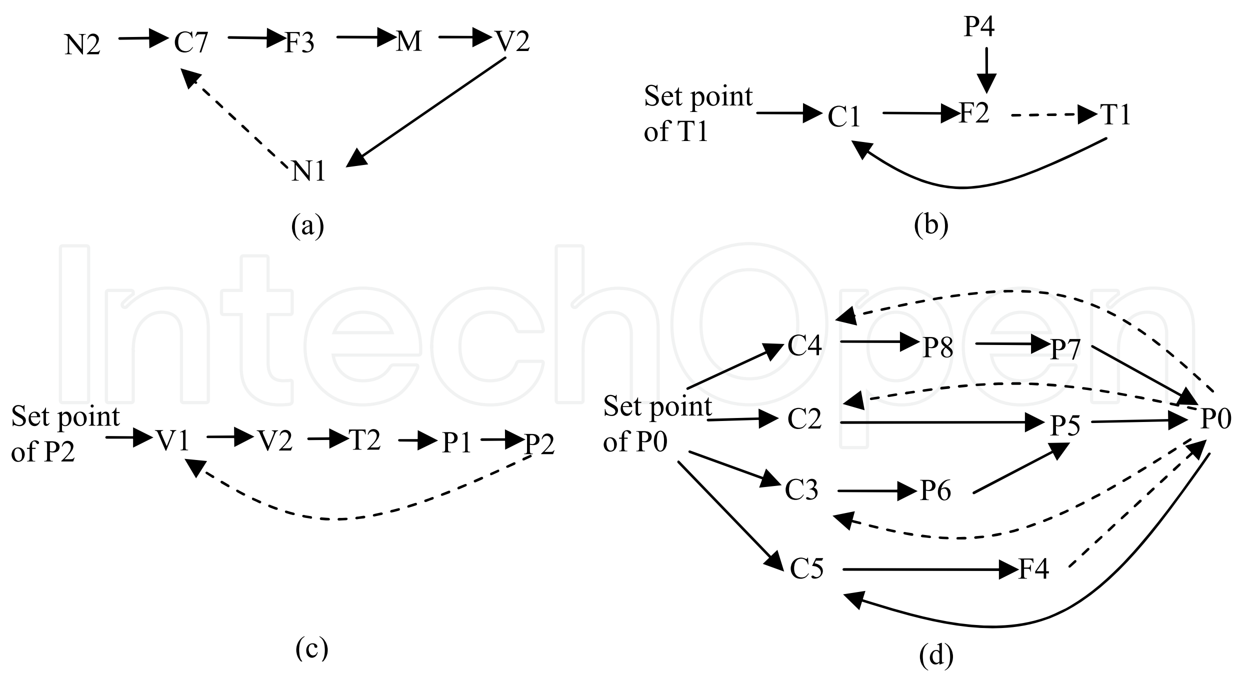

Fig. 21. Bottom-level SDGs of the generator set process. (a) S2, (b) S3, (c) S4, (d) S5

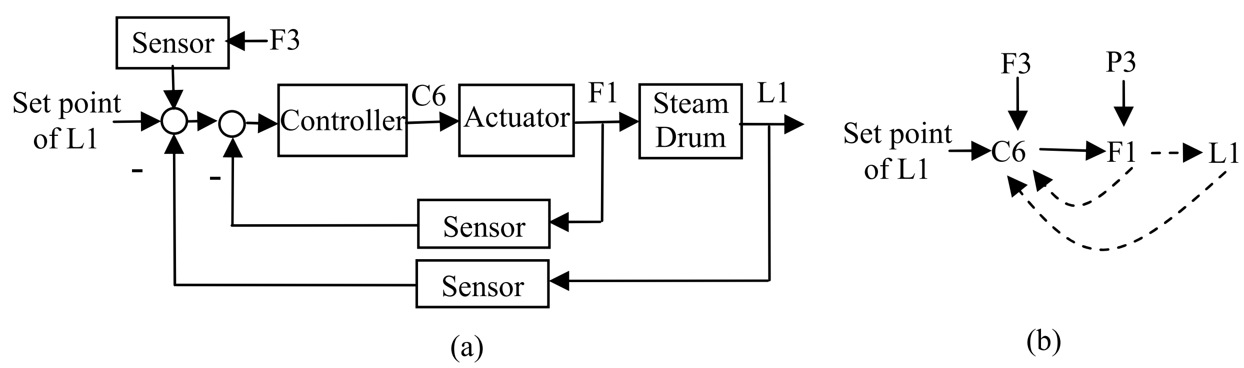

Fig. 22. Three-element water level control system. (a) Block diagram, (b) SDG

By combing the above sub-SDGs, the whole SDG of the generator process is shown as Fig. 23 where the red arcs stand for the control actions (Yang, 2008).

5.2 Fault Analysis of the Generator Set Process

When fault occurs, symptoms can be explained by SDG inference. Typical faults and their fault propagation paths are summarized as Table 6. Along the paths we can find the possible fault origins.

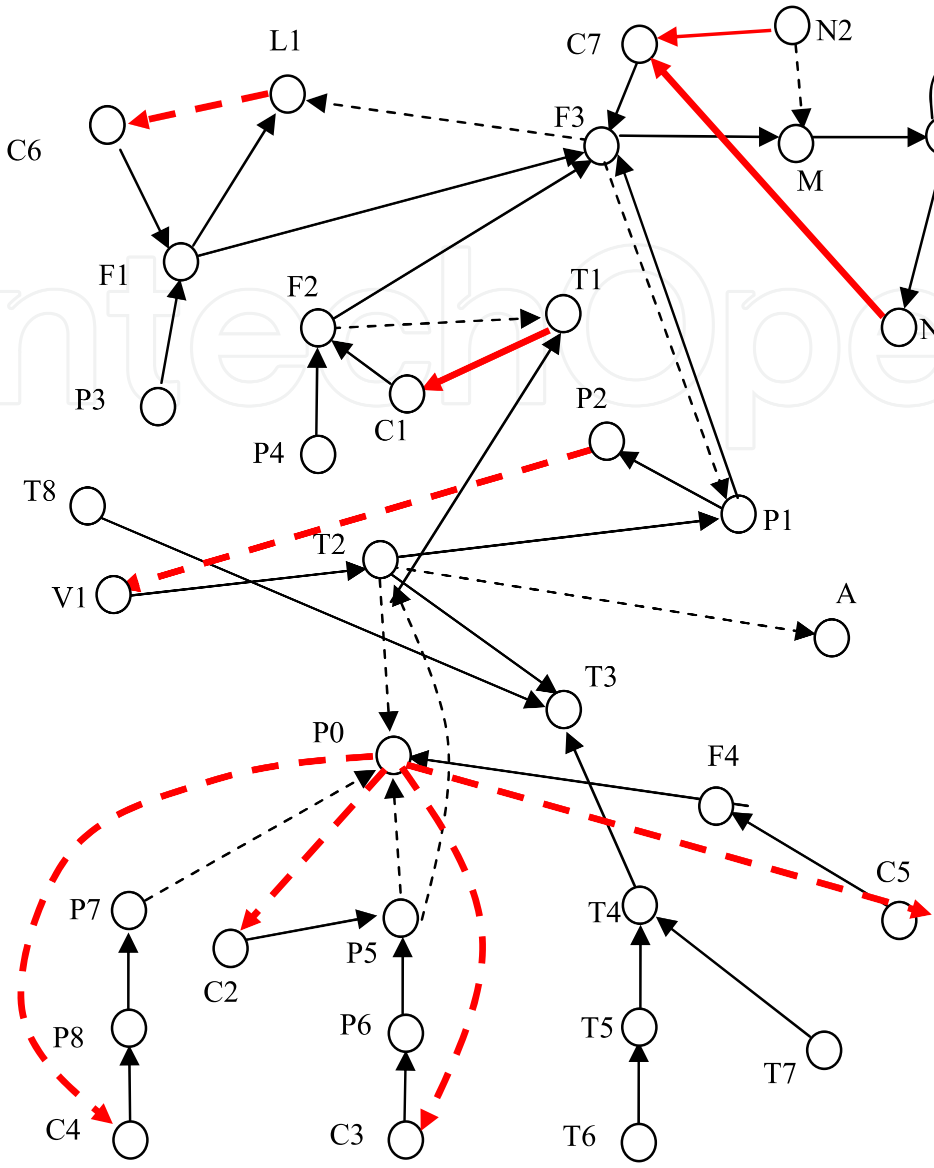

Fig. 23. Single-level SDG of the generator set process.

Fault	Fault propagation path in middle level	Fault propagation path in bottom level
Full of water in steam drum	S1	F1(+)→L1(+)
Lack of water in steam drum	S1	F1(-)→L1(-)→C6(+)
Too large of draught fan baffle	S5	C5(+)→F4(+)→P0(-)
Increase of load	S2	N2(+)→C7(+)→F3(+) M(-)→V2(-)→N1(-)→C7(+)
Change of coal quality	S4→S5 S4→T3 S4→S3→S2→S1	T2(+)→P1(+)→P2(+)

Table 6. Typical faults and their fault propagation paths

In the case of coal quality change, from the middle-level inference we find the fault origin is located in S4. Thus we just go on to make inference in the SDG of S4 and ignore other symptoms. If we make inference in a single-level SDG, then there are other paths (shown in Fig. 22) that are all redundant for fault origin search:

T2(+)→P0(+)

T2(+)→T3(+)

T2(+)→T1(+)→C1(+)→F2(+)→F3(+)→M(+)→V2(+)→N1(+)

T2(+)→T1(+)→C1(+)→F2(+)→F3(+)→L1(-)→C6(-)

However these paths are useful for hazard analysis. In Fig. 24, some control arcs are deleted compared with Fig. 23 because they are usually performed manually in the application. From the propagation path, we find the hearth temperature T2 is the key variable, so adding sensors on it can improve the fault detection reliability.

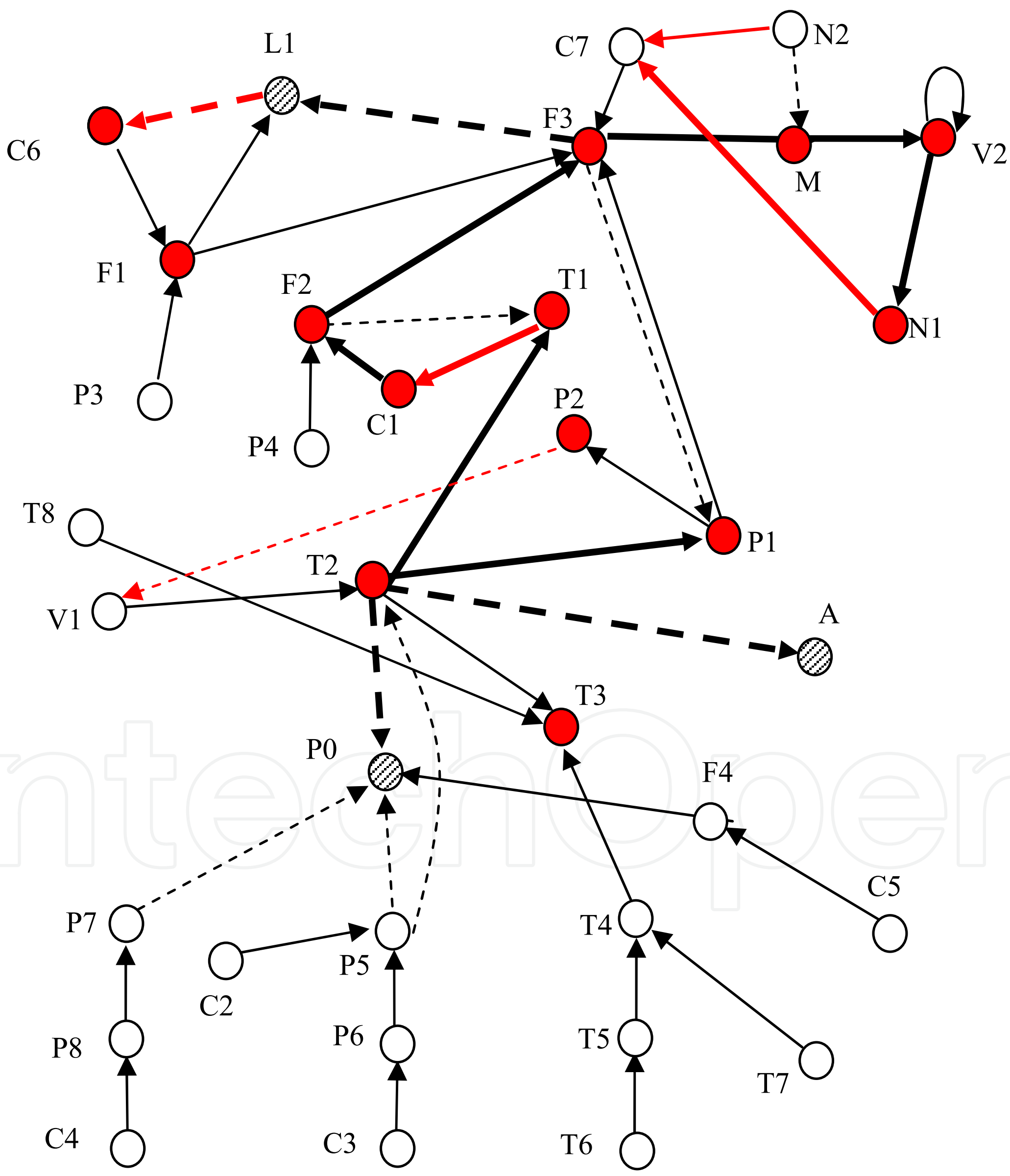

Fig. 24. Fault propagation when coal quality changes

6. Conclusion

In this chapter, after the introduction of the SDG concept and modeling methods, the inference approaches aiming at the fault detection and hazard analysis, especially the SDG description of control systems, have been analyzed from theory to practice.

The classical control methods on the basis of feedback idea are in common use, so the modeling and analysis of the systems under these control methods, have been discussed. When a control system is transformed into an SDG model, the direction of fault propagation in steady state may differ from the direction in initial response because of the control action. For a single-loop control system, the SDG is a directed path whose backbone is set point → measurement value → controlled variable → manipulated variable → controller output, which is also the fault propagation path and does not compose a loop. Based on this result, SDGs and fault propagation paths of various control systems can be obtained by the combination and connection of several single-loop control elements. Thus we do not have to list all the system equations when analyzing the actual problem, but only need to construct the local SDG for each separate control component and then combine them together, which is convenient for actual use. After analyzing the fault propagation paths in control systems, we can embed the resulted SDGs for initial response and final response into the SDG of the whole system and analyze the propagation paths considering the truncated or changed paths. This method enables the application of SDG method in large-scale complex systems.

It is to be noted that model description should meet the actual needs but does not need to be too accurate. For example, the SDG-based qualitative analysis of the three-element control of the boiler water level usually do not refer to the details, so usually we only construct a single loop to describe the major problem and ignore the minor ones. In large-scale complex systems, however, SDG models can be adopted to describe the interactions between different parts and reveal the propagation for the use of fault analysis; it is the advantage of SDG models.

In industrial systems, alarm monitoring design is a very important issue, among which the trade-off between missed alarms and false alarms should be treated appropriately. We should pay attention to two levels of design problems: In the local level, the threshold selection, data filtering and alarm triggering are the key problems to be solved. In the system level, topology expression and sensor location for alarm rationalization is essential. In this chapter, we have described and solved the sensor location problem aiming at the trade-off with the help of topology expressed by SDG. The optimization objective is expressed as the minimization of all the fault undetectabilities in the system. The false alarm rate is used as constraint as well as the cost limit.

Our future work may include: standard modeling method using XML-based process knowledge, modeling and validation of SDGs using process data, and combination of qualitative fault propagation and quantitative diagnosis.

7. Acknowledgment

The authors gratefully acknowledge the financial aid for this research by the National Natural Science Foundation of China (No. 60736026, 60904044), China Postdoctoral Science Foundation (No. 20080440386) and NSERC (Natural Sciences and Engineering Research Council of Canada) – Suncor – Matrikon – iCORE Industrial Research Chair Program.

8. References

Bauer, M.; Cox, J.W.; Caveness, M.H.; Downs, J. & Thornhill, N.F. (2007). Finding the direction of disturbance propagation in a chemical process using transfer entropy. *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 1, January 2007, pp. 12-21, ISSN: 1063-6536

Bhushan, M. & Rengaswamy, R. (2002). Comprehensive design of a sensor network for chemical plants based on various diagnosability and reliability criteria. 1. Framework. *Industrial & Engineering Chemistry Research*, Vol. 41, No. 7, April 2002, pp. 1826-1839, ISSN: 0888-5885

Chang, C.C. & Yu, C.C. (1990). On-line fault diagnosis using the signed directed graph. *Industrial & Engineering Chemistry Research*, Vol. 29, No. 7, July 1990, pp. 1290-1299, ISSN: 0888-5885

Chen, J. Y. & Chang, C.T. (2007). Systematic enumeration of fuzzy diagnosis rules for identifying multiple faults in chemical processes. *Industrial & Engineering Chemistry Research*, Vol. 46, No. 11, May 2007, pp. 3635-3655, ISSN: 0888-5885

Gentil, S. & Montmain, J. (2004). Hierarchical representation of complex systems for supporting human decision making. *Advanced Engineering Informatics*, Vol. 18, No. 3, July 2004, pp. 143–159, ISSN: 1474-0346

Graphviz (2009). URL: http://www.graphviz.org

Han, C.C.; Shih, R.F. & Lee, L.S. (1994). Quantifying signed directed graphs with the fuzzy set for fault diagnosis resolution improvement. *Industrial & Engineering Chemistry Research*, Vol. 33, No. 8, August 1994, pp. 1943-1954, ISSN: 0888-5885

Iri, M.; Aoki, K.; O'shima, E. & Matsuyama, H. (1979). An algorithm for diagnosis of system failures in the chemical process. *Computers & Chemical Engineering*, Vol. 3, No. 1-4, 1979, pp. 489–493, ISSN: 0098-1354

Iri, M; Aoki, K.; O'Shima, E. & Matsuyama, H. (1980). A graphical approach to the problem of locating the origin of the system failure. *Journal of the Operations Research Society of Japan*, Vol. 23, No. 4, December 1980, pp. 295–311, ISSN: 0453-4514

Izadi, I.; Shah, S.; Kondaveeti, S. & Chen, T. (2009). A framework for optimal design of alarm systems, *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Barcelona, Spain, July 2009

Jiang, H.; Patwardhan, R. & Shah, S. (2008). Root cause diagnosis of plant-wide oscillations using the adjacency matrix, *Proceedings of 17th IFAC World Congress*, pp. 13893-13900, ISBN: 978-3-902661-00-5, Seoul, Korea, July 2008, Elsevier, Amsterdam

Kramer, M.A. & Palowitch, B.L. (1987). A rule-based approach to fault diagnosis using the signed directed graph. *AIChE Journal*, Vol. 33, No. 7, July 1987, pp. 1607-1078, ISSN: 0001-1541

Lee, G.; Tosukhowong, T.; Lee, J.H. & Han, C. (2006). Fault diagnosis using the hybrid method of signed digraph and partial least squares with time delay: the pulp mill process. *Industrial & Engineering Chemistry Research*, 2006, Vol. 45, No. , pp. 9061-9074, ISSN: 0888-5885

Mah, R.S.H. (1989). *Chemical Process Structures and Information Flows*. Butterworth Publishes, ISBN: 0-409-90175-X, Boston

Maurya, M.R.; Rengaswamy, R. & Venkatasubramanian, V. (2003a). A systematic framework for the development and analysis of signed digraphs for chemical processes. 1. Algorithms and analysis. *Industrial & Engineering Chemistry Research*, Vol. 42, No. 20, October 2003, pp. 4789-4810, ISSN: 0888-5885

Maurya, M.R.; Rengaswamy, R. & Venkatasubramanian, V. (2003b). A systematic framework for the development and analysis of signed digraphs for chemical processes. 2. Control loops and flowsheet analysis. *Industrial & Engineering Chemistry Research*, Vol. 42, No. 20, October 2003, pp. 4811-4827, 2003, ISSN: 0888-5885

Maurya, M.R.; Rengaswamy, R. & Venkatasubramanian, V. (2006). A signed directed graph-based systematic framework for steady-state malfunction diagnosis inside control loops. *Chemical Engineering Science*, Vol. 61, No. 6, March 2006, pp. 1790-1810, ISSN: 0009-2509

McCoy, S.A.; Wakeman, S.J.; Larkin, F.D.; Chung, P.W.H. & Rushton, A.G. & Lees, F.P. (1999). HAZID, a computer aid for hazard identification. 1. The STOPHAZ package and the HAZID code: an overview, the issues and the structure. *Transactions of the Institution of Chemical Engineers. Part B*, Vol. 77, No. 6, November 1999, pp. 317-327, ISSN: 0957-5820

Mylaraswamy, D. & Venkatasubramanian, V. (1997). A hybrid framework for large scale process fault diagnosis. *Computer & Chemical Engineering*, Vol. 21, No. Suppl., May 1997, pp. S935-S940, ISSN: 0098-1354

Oyeleye, O. O. & Kramer, M. A. (1988). Qualitative simulation of chemical process systems: steady-state analysis. *AIChE Journal*, Vol. 34, No. 9, September 1988, pp. 1441-1454, ISSN: 0001-1541

Preisig, H. A. (2009). A graph-theory-based approach to the analysis of large-scale plants. *Computers & Chemical Engineering*, Vol. 33, No. 3, March 2009, pp. 598-904, ISSN: 0098-354

Raghuraj, R.; Bhushan, M. & Rengaswamy, R. (1999). Locating sensors in complex chemical plants based on fault diagnostic observability criteria. *AIChE Journal*, Vol. 45, No. 2, February 1999, pp. 310-322, ISSN: 0001-1541

Shiozaki, J.; Matsuyama, H.; O'Shima, E. & Iri, M. (1985). An improved algorithm for diagnosis of system failures in the chemical process. *Computers & Chemical Engineering*, Vol. 9, No. 3, pp. 285-293, ISSN: 0098-1354

Shiozaki, J.; Shibata, B.; Matsuyama, H. & O'shima, E. (1989). Fault diagnosis of chemical processes utilizing signed directed graphs-improvement by using temporal information. *IEEE Transactions on Industrial Electronics*, Vol. 36, No. 4, November 1989, pp. 469-474, ISSN: 0278-0046

Takeda, K.; Shibata, B.; Tsuge, Y. & Matsuyama, H. (1995). Improvement of fault diagnostic system utilizing signed directed graph–the method using transfer delay of failure. *Transactions of the Society of Instrument and Control Engineers*, Vol. 31, No. 1, January 1995, pp. 98–107, ISSN: 0453-4654

Thambirajah, J.; Benabbas, L.; Bauer, M. & Thornhill, N. F. (2009). Cause-and-effect analysis in chemical processes utilizing XML, plant connectivity and quantitative process history. *Computers & Chemical Engineering*, Vol. 33, No. 2, February 2009, pp. 503-512, ISSN: 0098-1354

Vedam, H. & Venkatasubramanian, V. (1997). Signed digraph based multiple fault diagnosis. *Computers & Chemical Engineering*, Vol. 21, No. Suppl., May 1997, pp. S655-S660, ISSN: 0098-1354

Vedam, H. & Venkatasubramanian, V. (1999). PCA-SDG based process monitoring and fault diagnosis. *Control Engineering Practice*, Vol. 7, No. 7, July 1999, pp. 903-917, ISSN: 0967-0661

Venkatasubramanian, V.; Rengaswamy, R. & Kavuri, S.N. (2003). A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, Vol. 27, No. 3, March 2003, pp. 313-326, ISSN: 0098-1354

Venkatasubramanian, V.; Zhao, J. & Viswanathan, S. (2000). Intelligent system for HAZOP analysis of complex process plants. *Computers & Chemical Engineering*, Vol. 24, No. 9-10, October 2000, pp. 2291-2302, ISSN: 0098-1354

Wilcox, N.A. & Himmelblau, D.M. (1994). The possible cause and effect graphs (PCEG) model for fault diagnosis—I. Methodology. *Computers & Chemical Engineering*, Vol. 18, No. 2, pp.103-116, February 1994, ISSN: 0098-1354

Yang, F. (2008). *Research on Dynamic Description and Inference Approaches in SDG Model-Based Fault Analysis [PhD Dissertation]*, Tsinghua University, Beijing

Yang, F. & Xiao, D. (2005a). Review of SDG modeling and its application. *Control Theory and Applications*, Vol. 22, No. 9, October 2005, pp. 767-774, ISSN: 1000-8152

Yang, F. & Xiao, D. (2005b). Approach to modeling of qualitative SDG model in large-scale complex systems. *Control and Instruments in Chemical Industry*, Vol. 32, No. 5, October 2005, pp. 8-11, ISSN: 1000-3932

Yang, F. & Xiao, D. (2006a). Approach to fault diagnosis using SDG based on fault revealing time, *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pp. 5744-5747, ISBN: 1-4244-0332-4, Dalian, China, June 2006, IEEE, New York

Yang, F. & Xiao, D. (2006b). Probabilistic signed directed graph and its application in hazard assessment, In: *Progress in Safety Science and Technology (Vol. VI)*, Huang, P.; Wang, Y.; Li, S.; Zheng, C. & Mao, Z. (Ed.), pp. 111-115, Science Press, ISBN: 7-03-018145-X, Beijing

Yang, F. & Xiao, D. (2006c). Model and fault inference with the framework of probabilistic SDG, *Proceedings of 9th International Conference on Control, Automation, Robotics and Vision*, pp. 1023-1028, ISBN: 1-4244-0341-3, Singapore, December 2006, IEEE, New York

Yang, F. & Xiao, D. (2007). Expressions of logic operators in signed directed graph and its applications in complex systems. *DCDIS Series B*, Vol. 14, No. S2, pp. 1260-1264, ISSN: 1492-8760

Yang, F. & Xiao, D. (2008a). SDG-based fault isolation for large-scale complex systems solved by rough set theory, *Proceedings of 17th IFAC World Congress*, pp. 7221-7226, ISBN: 978-3-902661-00-5, Seoul, Korea, July 2008

Yang, F. & Xiao, D. (2008b). Sensor location strategy in large-scale systems for fault detection applications. *Journal of Computers*, Vol. 3, No. 11, November 2008, pp. 51−57, ISSN: 1796-203X

Yu, C.C. & Lee, C. (1991). Fault diagnosis based on qualitative/quantitative process knowledge. *AIChE Journal*, Vol. 37, No. 4, April 1991, pp. 617-628, ISSN: 0001-1541

Zhang, Z.; Wu, C.; Zhang, B.; Xia, T. & Li, A. (2005). SDG multiple fault diagnosis by real-time inverse inference. *Reliability Engineering & System Safety*, Vol. 87, No. 2, May 2005, pp. 173-189, ISSN: 0951-8320

Fault Detection
Edited by Wei Zhang

ISBN 978-953-307-037-7
Hard cover, 504 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

In this book, a number of innovative fault diagnosis algorithms in recently years are introduced. These methods can detect failures of various types of system effectively, and with a relatively high significance.