J Intell Manuf (2007) 18:249–260
DOI 10.1007/s10845-007-0018-8

Failure diagnosis and recovery based on DES framework

Hyoung Il Son · Suk Lee

Received: November 2005 / Revised: May 2006 / Published online: April 2007

Abstract As many industrial systems become more complex, it becomes extremely difficult to diagnose the cause of failures. This paper presents a new failure diagnosis approach based on discrete-event systems (DES) framework. In particular, the approach is a hybrid of event-based and state-based ones leading to a simpler failure diagnoser with supervisory control capability. In our approach, we include the failure recovery events for failures in the system model in order to derive a diagnoser we refer to as a recoverable diagnoser. Further, in order to reduce the state size of the recoverable diagnoser, a procedure to construct a high-level diagnoser is presented. The design procedure for diagnoser is presented along with a pump-valve system as a illustrative example.

Keywords Failure diagnosis · Failure recovery · Discrete-event systems · Model reduction

H. Il Son
Korea Advanced Institute of Science and Technology,
Daejeon 305-701, Korea

Present address:
H. Il Son (✉)
Information Technology Display Center, Samsung
Electronics, Chungcheonganm-do 330-300, Korea
e-mail: hyoungil.Son@samsung.com

S. Lee
School of Mechanical Engineering, Pusan National
University, Pusan 609-735, Korea
e-mail: slee@hyowon.pusan.ac.kr

Introduction

Failure diagnosis in industrial system is a subject that received a great deal of attention in the past few decades. To solve diagnostic problems for large complex systems such as semiconductor manufacturing systems, automobile manufacturing systems, chemical processes, Heating, Ventilation and Air Conditioning (HVAC) units, and power plants, more systematic and efficient approaches are required because of the tremendous problem sizes.

In order to develop practical diagnostic systems, many theoretical frameworks have been proposed. These include fault tree analysis, analytical redundancy, expert systems, model-based reasoning methods. Even though these techniques have their own merits, the real-world applications require some ways to circumvent individual limitations such as prohibitive computational burden, excessive sensitivity to modeling errors and sensor noise, and lack of systematic knowledge acquisition. Recently, DES methods (Brandt et al., 1990; Cassandras, 1993; Davey & Priestley, 1990; Hopcroft & Ullman, 1979; Kumar & Garg, 1995; Ramadge & Wonham, 1989; Wonham, 1998; Wonham & Ramadge, 1987) are recognized as one of the promising techniques because most industrial systems is better modeled by a discrete-event model than by a differential or difference equation model at a higher level of abstraction (Lin & Lin, 1993). And DES technique can offer more systematic and simpler construction of a diagnostic system. In a discrete-event model, normal and failed status of system components are represented by states and control commands, sensor signals and failure events form the event set. The failure diagnosis problem is to detect the occurrence of failure events that is not observable by using normal events, e.g. control commands and sensor signals that is observable. The major advantage of this approach is that is does not require detailed in-depth modeling of the system to be diagnosed and hence is ideally suited for diagnosis of systems which are difficult to model (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). Due to these advantages, there have been many approaches to the diagnosis problem by using DES technique (Hashtrudi Zad, 1999; Hashtrudi Zad, Kwong & Wonham, 2003; Lin & Lin, 1993; Park, 1996; Sampath, 1995; Sampath, Lafortune, & Teneketzis, 1998; Sampath et al., 1995).

Failure diagnosis system can be classified by two criteria. The first criterion is what is the state of a system when the diagnostic procedure is applied. In the off-line failure diagnosis, the system is assumed to be in an abnormal state when the failure diagnosis beings. The diagnostic system collects information from the failed system to draw inferences on the state of system and the cause of the failure. In contrast with the off-line failure diagnosis, the on-line failure diagnosis applied while the system is in normal operation. The on-line diagnostic system collects information to determine whether the system is normal or not. If the system is in abnormal states, the diagnostic procedure tries to find the cause. The second criterion for classifying diagnosis system is whether the failure diagnoser actively intervenes with the system's operation. The passive failure diagnosis does not affect the system's operation. Instead, it simply observes the sequence of events and keeps the tract of system states. On the other hand, the active failure diagnosis can change the system's operation by issuing a sequence of commands to determine the system's state and the cause of the failure (Hashtrudi Zad, 1999; Lin & Lin, 1993; Sampath et al., 1998).

The DES approach to the diagnosis problem is divided into an event-based approach and state-based approach (Hashtrudi Zad et al., 2003). In general, the event-based approach is simpler in its design procedure than the state-based approach. But its drawback is that usually the designed diagnoser has more states than the other. And, unlike the event-based diagnosis, state-based approach requires the knowledge on the state of the supervisor that controls the system's behavior. This means that the diagnosis operation have to be synchronized with the operation of supervisor. This paper present an approach to design a passive on-line diagnoser based on the DES framework. Unlike the existing DES techniques, this approach is a hybrid of the event-based and the state-based approaches leading to a simpler failure diagnoser than event-based approached one with supervisory control capability not required the synchronization with supervisor. Furthermore we introduce the concept of recoverability by taking recovery events for failures into consideration. By this extension, the diagnoser can allow the system to recover from a failure as well as detect and isolate the failure. Also, a procedure to construct a high-level diagnoser is presented in order to reduce the state size of the diagnoser.

This paper is organized in five sections. In the section following this introduction, a DES modeling technique and a design procedure for the proposed diagnoser are presented. In the "Diagnosability and recoverability" section, the concept of diagnosability along with its the necessary and sufficient conditions is presented. In addition, the concept of recoverability that is first introduced by this paper is defined with its the necessary and sufficient conditions. The methodology for reducing the state size of the recoverable diagnoser is presented in the "High-level diagnoser design" section. As an illustrative example, a simple pump-valve system is used throughout the paper. Finally, we summarize the main contributions of this paper and outline the directions for future research in the "Conclusion" section.

Recoverable diagnoser design

In this section we will present the DES modeling procedure and the diagnoser design procedure based on the DES framework. While existing DES methods for failure diagnosis do not deal the failure recovery problem, our approach take into account failure recovery events in the DES modeling step. By this approach diagnoser makes a DES return to the initial normal by enabling a failure recovery event state when the diagnoser detects the failure event.

DES modeling

In general, we can assume that the DES to be diagnosed has a several system components including the plant components to be controlled and the supervisor for control action. First, let these plant components be modeled by the Finite State Automata (FSA):

$$\overline{G_i} = \left\{\overline{Q_i}, \overline{\Sigma_i}, \overline{\delta_i}, \overline{q_{0,i}}, \overline{Q_{m,i}}\right\}, \quad i = 1, \ldots, n \qquad (1)$$

where $\overline{Q_i}$ is the state set, $\overline{\Sigma_i}$ is the event set, $\overline{\delta_i} : \overline{Q_i} \times \overline{\Sigma_i}^* \to \overline{Q_i}$ is the sate transition function , $\overline{q_{0,i}}$ is the initial state, $\overline{Q_{m,i}}$ is the marked state set that is a subset of the state set $\overline{Q_i}$. In defining transition function $\overline{\delta_i}$, the notation $\overline{\Sigma_i}^*$ means the set of sequences (strings) of events including the null event ε. To define the failure events of each component let us define the failure event set of a component as $\overline{\Sigma_{F,i}} = \left\{f_i^1, f_i^2, \ldots, f_i^m\right\}$. And define the failure recovery event set for failure event as $\overline{\Sigma_{RF,i}} = \left\{RF_i^1, Rf_i^2, \ldots, Rf_i^n\right\}$, $n \leq m$ (assuming that some failure recovery event can take care of more than one failure events). So the normal event set is defined as $\overline{\Sigma_{N,i}} = \overline{\Sigma_i} - \overline{\Sigma_{F,i}} - \overline{\Sigma_{RF,i}}$. As a result we partition the event set $\overline{\Sigma_i}$ into three disjoint event sets, i.e., the normal event set $\overline{\Sigma_{N,i}}$, the failure event set $\overline{\Sigma_{F,i}}$, and failure recovery event set $\overline{\Sigma_{RF,i}}$. That is, $\overline{\Sigma_i} = \overline{\Sigma_{N,i}} \dot{\cup} \overline{\Sigma_{F,i}} \dot{\cup} \overline{\Sigma_{RF,i}}$ where $\dot{\cup}$ denotes a disjoint union. Then, we can obtain the FSA of the total plant by the synchronous product of all FSAs $\overline{G_i}$. The resulting FSA is denoted by

$$\overline{G} = \left\{\overline{Q}, \overline{\Sigma}, \overline{\delta}, \overline{q_0}, \overline{Q_m}\right\} \tag{2}$$

where $\overline{Q}$, $\overline{\Sigma}$, $\overline{\delta}$, $\overline{q_0}$, and $\overline{Q_{m,i}}$ follow the previous definitions. In particular, the event set $\overline{\Sigma}$ can be divided into two disjoint sets, i.e., the controllable event set $\overline{\Sigma_{\mathrm{c}}}$ and the uncontrollable event set $\overline{\Sigma_{\mathrm{uc}}}$. And $\overline{\Sigma}$ can be also partitioned into the observable event set $\overline{\Sigma_{\mathrm{o}}}$ and the unobservable event set $\overline{\Sigma_{\mathrm{uo}}}$ that are also disjoint. Therefore, $\overline{\Sigma}$ can be written as $\overline{\Sigma} = \overline{\Sigma_{\mathrm{c}}} \dot{\cup} \overline{\Sigma_{\mathrm{uc}}} = \overline{\Sigma_{\mathrm{o}}} \dot{\cup} \overline{\Sigma_{\mathrm{uo}}}$.

As the supervisor for $\overline{G}$, $\dot{\cup}$ can be used to generate the supremal controllable and observable sublanguage where S is an FSA, $S = \{X, \Sigma, \xi, x_0, X_m\}$ that can be obtained by the results in Brandt et al. (1990), Wonham (1998) and φ is a control map defined as $\varphi : X \to 2^{\Sigma} (\supseteq \Sigma_{uc})$. With $\overline{G}$ as the total plant and the supervisor (S, φ), the total system can be represented by a Finite State Moore Automaton (FSMA)[1]:

$$G = \{Q, \Sigma, \delta, q_0, Q_m, Y, \lambda, C, \gamma\} \tag{3}$$

that is obtained by the meet product of plant FSA $\overline{G}$ and supervisor FSA S. In the FSMA G, $Q = \overline{Q} \times X$ is the state set; $\Sigma \subseteq \overline{\Sigma}$ is the event set; $\delta : Q \times \Sigma \to 2^Q$ is the transition function; $q_0 = (\overline{q_0}, x_0)$ is the initial state with $\overline{q_0}$ and x_0 are the initial states of the plant and the supervisor, respectively; and Q_m is the marked state set that is a subset of the state set Q. Among the new components, Y is the sensor output set, $\lambda : Q \to Y$ is the sensor output map, $C \subseteq \Sigma_c$ is the control command set, and $\gamma : Q \to 2^C$ is the control command map. Here, the sensor output means the results of sensor measurements of the system while the control command set includes the event can be enabled by the supervisor.

Example 2.1 (*Pump-valve system—system modeling and supervisor*) Consider a simple system consisting of a pump, a valve and a supervisor is illustrated in Fig. 1. And assume that we use a flowmeter as a measurement sensor. The value of the flowmeter can take two values, i.e., NF, F meaning noflow and flow, respectively. Thus, the sensor output set is $Y = \{\mathrm{NF}, \mathrm{F}\}$.

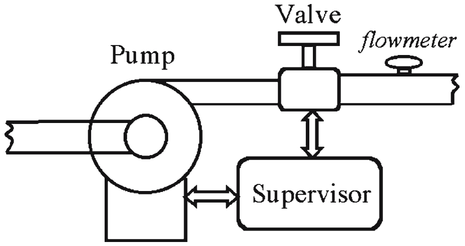

Fig. 1 Pump-valve system

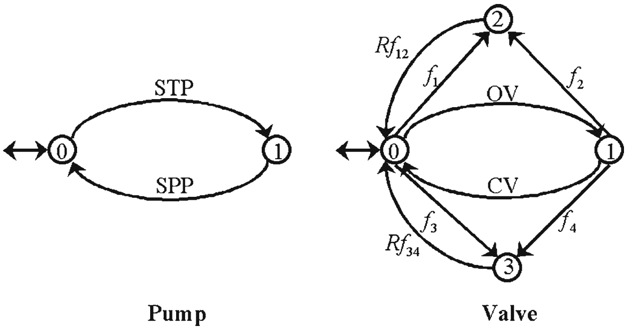

Fig. 2 Components modeling of pump-valve system

For simplicity, also assume that there are no failures in the pump, and only the valve has failures that are Stuck_closed1, Stuck_closed2, Stuck_open1, and Stuck_open2 denoted by f_1, f_2, f_3, and f_4, respectively. Both Stuck_closed1 and Stuck_closed2 block the flow through the valve regardless of current state of the valve. Stuck_closed1 occurs when the valve is in its closed position while Stuck_closed2 occurs when the valve is open. Similarly, both Stuck_open1 (when the valve is closed) and Stuck_open2 (when the valve is open) make the valve unable to stop the flow. Then, we define the failure recovery event for Stuck_closed1 and Stuck_closed2 as Rf_{12} because these will cause the same problem and repair operation can be very similar. In parallel, Rf_{34} is defined to be the failure recovery event for Stuck_open1 and Stuck_open2.

The FSAs of the pump and the valve are shown in Fig. 2. The initial state and marked state of the valve and pump are identical and represented by state 0. In Fig. 2 events OV, CV, STP, and STP stand for Open_valve, Close_valve, Start_pump, and Stop_pump, respectively. The uncontrollable event set is $\overline{\Sigma_{\mathrm{uc}}} = \{f_1,\ f_2,\ f_3,\ f_4\}$ and it is assumed that $\overline{\Sigma_{\mathrm{uc}}} = \overline{\Sigma_{\mathrm{uo}}}$ because failure events are unobservable events. The legal language for the system is represented in Fig. 3. Now we define the control command set as $C = \{\mathrm{OV}, \mathrm{CV}, \mathrm{STP}, \mathrm{SPP}, Rf_{12}, Rf_{34}\}$.

Next, the supervisor S that can generate the supremal controllable and observable sublanguage with respect

[1] In general, $L(S/\overline{G}) = L(S)$ is true (Kumar & Garg, 1995; Ramadge & Wonham, 1989; Sampath et al., 1998) so we can obtain G by just adding λ, Y to S.

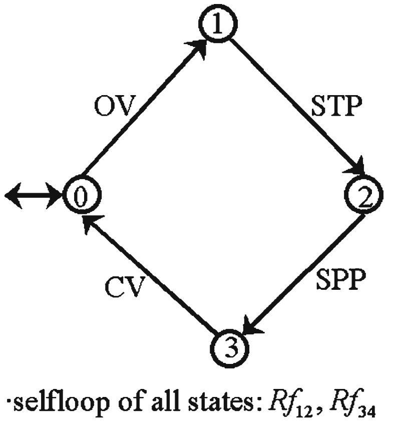

Fig. 3 Legal behavior FSA of pump-valve system

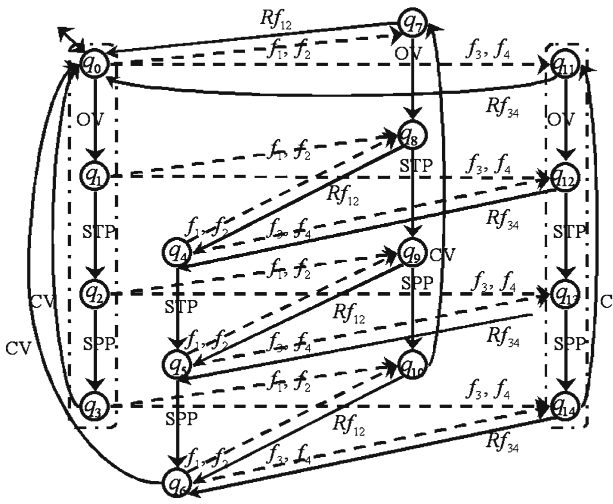

Fig. 4 FSMA of pump-valve system

to legal language is obtained (Brandt et al., 1990). The resulting FSMA G is depicted in Fig. 4. The transition caused by an observable event is shown by a solid line while a dashed line represents an unobservable failure event.

Table 1 shows the sensor output map λ, i.e., the sensor measurements at each state. In general, the sensor output changes from NF to F only after the pump is started following the valve opening. Note that the sensor output of q_0 is NF because of the failure event Stuck_closed1 or Stuck_closed2, which means that, the valve is still closed. Table 2 shows the control command λ, i.e., the list of eligible[2] events (also controllable events) at each state.

Recoverable diagnoser

As the first step to develop a recoverable diagnoser, the event set Σ is partitioned into the normal event set Σ_N, the failure event set Σ_F and the failure recovery event set Σ_{RF}.[3] That is, $\Sigma = \Sigma_N \dot\cup \Sigma_F \dot\cup \Sigma_{RF}$. And assume the failure event set Σ_F can be defined as $\Sigma_F = \{f_1, f_2, \ldots, f_m\}$ for a failure event $f_i, i = 1, 2, \ldots, m$. Denote the state q' that is reached from a certain state q of G by the failure event f_i as q_{f_i}: namely, q_{f_i} is defined as $q_{f_i} = q' = \delta(q, f_i)$. We define $F_1, F_2, \ldots, F_n, n \leq m$ as failure modes, also $K = \{N, F_1, F_2, \ldots., \quad F_n\}, n \leq m$ as state condition set of DES. Failure modes partition the failure event set Σ_F into n groups by aggregating some failure events that can be considered identical. In addition, assume that there is F_i-recovery event that is denoted by RF_i for each failure mode F_i. So we can define the failure recovery event set as $\Sigma_{RF} = \{RF_1, RF_2, \ldots., RF_n\}, n \leq m$. Naturally all members of Σ_{RF} are controllable event, therefore $\Sigma_{RF} \subseteq \Sigma_{\mathrm{c}}$.

With the partitions on the event set, now let us define Q_N and Q_{F_i} by the following equations as disjoint partitions of the state set Q, i.e., $Q = Q_N \dot\cup Q_{F_1} \dot\cup Q_{F_2} \dot\cup \ldots \dot\cup Q_{F_n}$.

$$Q_N = q_0 \cup \left\{ q' \,\middle|\, \forall q \in Q_N, \forall \sigma \in \Sigma_N, q' = \delta(q, \sigma) \right\} \tag{4}$$

$$\begin{aligned} Q_{F_i} = &\left\{ q_{f_i} \,\middle|\, \forall q \in Q_N, \forall f_i \in F_i, q_{f_i} = \delta(q, f_i) \right\} \\ &\cup \left\{ q' \,\middle|\, \forall q \in Q_{F_i}, \forall \sigma \in \Sigma_N, q' = \delta(q, \sigma) \right\} \end{aligned} \tag{5}$$

Finally, by the state condition and disjoint state sets defined in Eqs. (4) and (5), we define the state condition map $\kappa : Q \to K$ as follows

$$\kappa(q) = \begin{cases} N \text{ if } q \in Q_N \\ F_i \text{ if } q \in Q_{F_i} \end{cases} \tag{6}$$

Now let us define the recoverable diagnoser for DES G as

$$D = \left\{ Z, E, \varsigma, z_0, Z_m, \overline{K}, \overline{\kappa} \right\} \tag{7}$$

where $Z = 2^Q - \phi$ is the state set, $E = Y \times C$ is the event set, z_0 is the initial state that is defined as $z_0 = \{q_0 \cup q_0'\}$, $Z_m \supseteq Q_m$ is the marked state set, $\overline{K} = 2^K - \phi$ is the state condition set, $\overline{\kappa} : Z \to \overline{K}$ is the state condition map. This type of diagnoser D is also referred to as *recoverable diagnostic supervisor* because it preserves the control map of the supervisor while performing diagnosis. For initial state definition, the notation q_0' is defined as follows and illustrated in Fig. 5.

$$q_0' = \bigcup_i \left\{ q' \,\middle|\, \forall f_i \in \Sigma_F, q' = \delta(q_0, f_i) \right\} \tag{8}$$

[2] Eligible event means a candidate can be enabled (actually occurred) by the supervisor.

[3] $\Sigma_N = \bigcup_i \overline{\Sigma_{N,i}}$, $\Sigma_F = \bigcup_i \overline{\Sigma_{F,i}}$, $\Sigma_{RF} = \bigcup_i \overline{\Sigma_{RF,i}}$.

Table 1 Sensor output map λ of pump-valve system

State	Sensor output						
q_0	NF			q_7	NF	q_{11}	NF
q_1	NF	q_4	NF	q_0	NF	q_{12}	NF
q_2	F	q_5	F	q_0	NF	q_{13}	F
q_3	NF	q_6	NF	q_{10}	NF	q_{14}	NF

Table 2 Control command map γ of pump-valve system

State	Control command						
q_0	OV			q_7	OV, Rf_{12}	q_{11}	OV, Rf_{34}
q_1	STP	q_4	STP	q_0	STP, Rf_{12}	q_{12}	STP, Rf_{34}
q_2	SPP	q_5	SPP	q_0	SPP, Rf_{12}	q_{13}	SPP, Rf_{34}
q_3	CV	q_6	CV	q_{10}	CV, Rf_{12}	q_{14}	CV, Rf_{34}

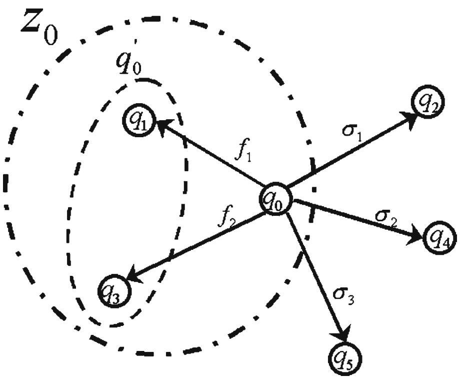

Fig. 5 Initial state of the recoverable diagnoser

This definition allows the diagnoser to start from either a normal or a faulty state. This is very advantageous in the sense that there is no problem in operation of the diagnoser no matter when a failure occurs, i.e., before or after the initialization of diagnoser. Some previous works assumed that the system starts from a normal state (Sampath, 1995; Sampath et al., 1995, 1998), which may not be general enough for many applications. And in Hashtrudi Zad (1999), the initial state is defined as all system states or all normal states. But this may be impractical in combining supervisory control and diagnosis because the supervisor has to start from some initial state.

In order to design the diagnoser defined in (7) from FSMA G, we have to construct a new transition system excluding all unobservable events such as failure events and the observable events that are not the member of control command set of the supervisor. The next definition defines this new transition system.

Definition 1 Define the Control Command Transition Systems (CCTS) as all transition (q, σ, q) of DES G such that

$$\forall \sigma \in C, q = \delta(q, \sigma)$$

CCTS contains only transitions that can be enabled by the supervisor. Physically, only events that are triggered by the control command from supervisor are contained in CCTS. So, CCTS are the subset of transitions that are monitored by the supervisor.

We define the transition function ς of the diagnoser D as follows

$$\varsigma(z_k, \gamma_k) = \begin{cases} z_{k+1} & \text{if } n(y_{k+1}) = 1 \\ z_{k+1+i}, 0 \leq i \leq m-1 & \begin{array}{l} \text{if } n(y_{k+1}) \geq 2, \\ y_{k+1} = \left\{ y_{k+1}^{0}, \ldots, y_{k+1}^{m-1} \right\} \end{array} \\ \text{undefined} & \text{elsewhere} \end{cases} \tag{9}$$

where $n(\bullet)$ means the number of $\bullet$.

The meaning of Eq. (9) can be explained by the following. If the supervisor issues a control command γ_k at the state z_k of diagnoser, then the state z_k transits to z_{k+1}. However, if there are more than one sensor outputs after the control command γ_k, we need to differentiate the next state z_{k+1} according to the sensor output. For this purpose, we define additional states as shown in Fig. 6.

Although this approach may look similar to Hashtrudi Zad (1999), Sampath (1995), Sampath et al. (1995), and Sampath et al. (1998), the transition function ς is different from the previous researches and can be more efficient. This is because of the following two reasons. First, the diagnoser in Sampath (1995), Sampath et al. (1995), and Sampath et al. (1998) has to update the state condition whenever any observable events occur while this approaches update the state condition only when there are changes in the value of sensor output and the supervisor issues the new control command. So the number of states of the new diagnoser is at most the same as that of the diagnoser presented in Sampath (1995), Sampath et al. (1995), and Sampath et al. (1998). The second

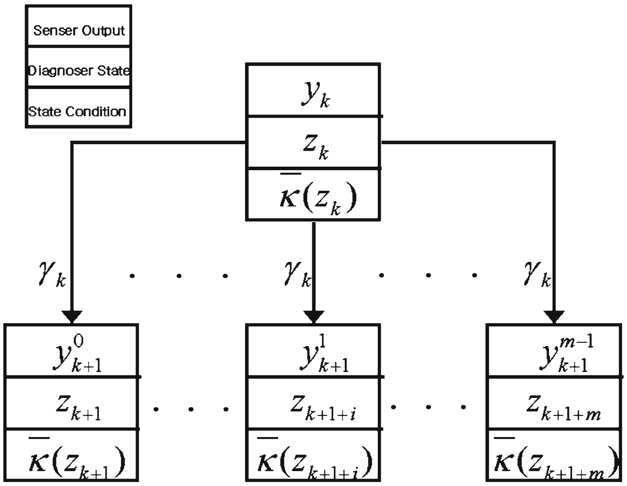

Fig. 6 Transition function ς

reason is related to the fact the state of the supervisor has to be included in the state output of the diagnoser presented in Hashtrudi Zad (1999). This requires the diagnoser to synchronize with the supervisor. In contrast, the diagnoser in this research includes both diagnostic and control capabilities, which implies that the recoverable diagnostic supervisor needs no synchronization and is much simpler than the diagnoser presented in Hashtrudi Zad (1999).

Figure 7 shows the schematic of on-line diagnosis for a DES. In Fig. 7, when the control command γ_k is issued at the present state z_k of the diagnoser, the state transits to a new state z_{k+1}. Then, the diagnoser reads the sensor output y_{k+1} of the present state z_{k+1}, and estimates the state condition of z_{k+1}.

In summary, the recoverable diagnoser design procedure is presented in the following.

Recoverable diagnoser design procedure:

Step 1: Define the failure modes from the failure events.
Step 2: Define the failure recovery event set Σ_{RF}.
Step 3: Define the state condition map κ by using (4), (5), and (6).
Step 4: Find the initial state of diagnoser by using (8).
Step 5: Construct CCTS.
Step 6: Define the transition function ς by using CCTS and (9).
Step 7: Build the recoverable diagnoser by using the transition function.

Example 2.2 (*Pump-valve system—recoverable diagnoser*) For the DES G constructed in Example 2.1, we can let the failure events f_1 and f_2 be grouped into failure mode F_1 because Stuck_closed1 and Stuck_closed2 are similar failure event. By the same reason, let the f_3 and f_4 be included in failure mode F_2. Then, failure recovery events Rf_{12} and Rf_{34} will be denoted as RF_1 and RF_2, respectively. Finally, the state condition set is $K = \{N, F_1, F_2\}$. From Fig. 4, we can easily define the state condition map κ shown in Table 3.

Before designing the diagnoser we can define the initial state z_0 of the diagnoser using (8). The initial state is $z_0 = \{q_0, q_7, q_{11}\}$ because $q_7 = \delta(q_0, F_1)$ and $q_{11} = \delta(q_0, F_2)$. And the CCTS for the DES G by definition 1 can be obtained by using Fig. 4, Tables 1 and 2. The result is listed in Table 4.

Table 4 is obtained by the following steps. Transitions that can occur in state q_0 are (q_0, OV, q_1), (q_0, F_1, q_7), and (q_0, F_2, q_{11}). However, the events F_1 and F_2 are not included in control command set C, therefore, the transitions (q_0, F_1, q_7) and (q_0, F_2, q_{11}) have to be excluded from CCTS. Instead, the transitions (q_0, OV, q_8), and (q_0, OV, q_{12}) have to included in CCTS because the diagnoser cannot tell the difference among the three states q_0, q_7, and q_{11}, and can only observe the transitions to q_8 and q_{12} from q_0 by the event of OV. Similarly other normal states, i.e., q_1, q_2, q_3, q_4, q_5, and q_6, the failure events are not observable from the diagnoser and the transition from these states can result in three different states as listed in Table 4.

By the CCTS listed in Table 4 and the transition function ζ defined by Eq. (9), the part of procedure for designing the diagnoser of the pump-valve system is illustrated in Fig. 8. If the control command OV is issued by the supervisor at the state $z_0 = \{q_0, q_7, q_{11}\}$, the state changes to states q_1, q_8, and q_{12} by the transition function δ defined in DES G. We can easily confirm this from Table 4. Also, because $\lambda(q_1) = \lambda(q_8) = \lambda(q_{12}) =$ NF, the state z_1 is defined as $z_1 = \{q_1, q_8, q_{12}\}$ by (9). If the control command STP is issued at the state $z_1 = \{q_1, q_8, q_{12}\}$ of diagnoser, the state changes to q_2, q_9, and q_{13}. But, in this case there are two sensor outputs, i.e., $\lambda(q_2) = \lambda(q_{13}) =$ F and $\lambda(q_9) =$ NF, so we have to define the states of the diagnoser as $z_2 = \{q_2, q_{13}\}$ and $z_3 = \{q_9\}$. By this design procedure, the recoverable diagnoser for the pump-valve system is built and shown in Fig. 9. In Fig. 9, the transition caused by control command is shown by a solid line while that caused by a failure recovery event is shown by a dashed line.

Diagnosability and recoverability

This section presents the definition of diagnosability and its necessary and sufficient conditions. In addition, the concept of recoverability is defined along with its necessary and sufficient conditions.

Fig. 7 On-line diagnosis

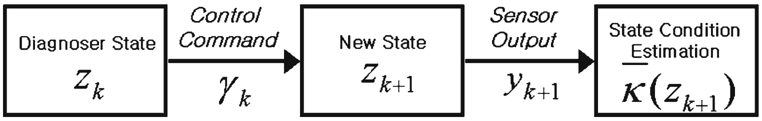

Table 3 State condition map κ of pump-valve system

State	State condition						
q_0	N			q_7	F_1	q_{11}	F_2
q_1	N	q_4	N	q_0	F_1	q_{12}	F_2
q_2	N	q_5	N	q_0	F_1	q_{13}	F_2
q_3	N	q_6	N	q_{10}	F_1	q_{14}	F_2

Table 4 CCTS of pump-valve system

State q	Control command $\sigma = \gamma(q)$	Destination states $q' = \delta(q, \sigma)$	Sensor outputs $\lambda(q)$
q_0	OV	q_1, q_0, q_{12}	NF
q_1	STP	q_2, q_{13}	F
		q_0	NF
q_2	SPP	q_3, q_{10}, q_{14}	NF
q_3	CV	q_0, q_7, q_{11}	NF
q_4	STP	q_5, q_{13}	F
		q_0	NF
q_5	SPP	q_6, q_{10}, q_{14}	NF
q_6	CV	q_0, q_7, q_{11}	NF
q_7	OV	q_0	NF
	RF_1	q_0	NF
q_0	STP	q_0	NF
	RF_1	q_4	NF
q_0	SPP	q_{10}	NF
	RF_1	q_5	F
q_{10}	CV	q_7	NF
	RF_1	q_6	NF
q_{11}	OV	q_{12}	NF
	RF_2	q_0	NF
q_{12}	STP	q_{13}	F
	RF_2	q_4	NF
q_{13}	SPP	q_{14}	NF
	RF_2	q_5	F
q_{14}	CV	q_{11}	NF
	RF_2	q_6	NF

Diagnosability

In this section we explain how the diagnoser estimates the state condition of DES and detects the failure event. For this purpose, the state set Z of the diagnoser is classified into normal, F_i-uncertain and F_i-certain. The precise definitions are provided in the following.

Definition 2 The state z of the diagnoser is said to be *normal* if

$$\overline{\kappa(z)} = \{N\}$$

Definition 3 The state z of the diagnoser is said to be F_i *-uncertain* if

$$\overline{\kappa(z)} \supset \{F_i\}, \overline{\kappa(z)} \not\subset \{F_i\}$$

Definition 4 The state z of the diagnoser is said to be F_i *-certain* if

$$\overline{\kappa(z)} = \{F_i\}$$

Based on the above definitions, diagnosability of a diagnoser is defined by the following.

Definition 5 A diagnoser is F_i *-diagnosable* if the state of the diagnoser can be F_i-certain after the occurrence

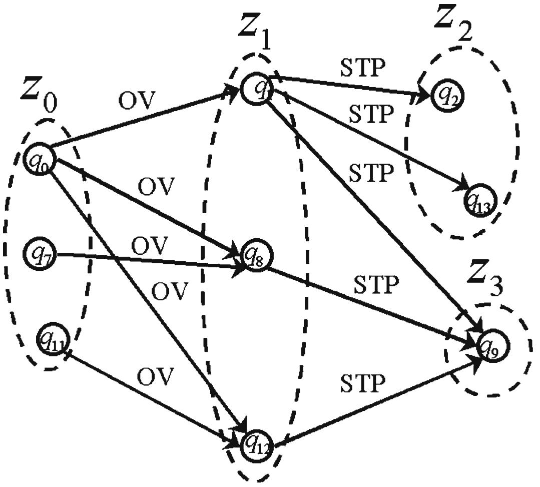

Fig. 8 A part of diagnoser design procedure

of at most a finite number of events, N_i, in the system, following both the initialization of the diagnoser and the occurrence of failure mode F_i. Also if diagnoser is F_i-diagnosable for all failure mode F_i then the diagnoser is said to be *diagnosable*.

The physical meaning of diagnosability can be explained by the following. Suppose that a system looks normal to the diagnoser even after a failure mode F_i occurred. In that case, the state condition of diagnoser contain not only F_i but also N, i.e. the state of the diagnoser is F_i-uncertain. However, if the state conditions of diagnoser contain only F_i without N after occurrences of some abnormal events, i.e. the state of the diagnoser is F_i-certain, the diagnoser can determine that the failure mode F_i has occurred without any ambiguity.

Example 3.1 (*Pump-valve system—diagnosability*) The states z_3, z_5, z_6, and z_7 of the diagnoser are F_1-certain as shown in Fig. 9. So the diagnoser for the pump-valve system is F_1-diagnosable by the Definition 5. But, the diagnoser is not F_2-diagnosable because there is no F_2-certain state.

Before we state the theorem for diagnosability, let us define a cycle and an indeterminate cycle.

Definition 6 A set of states $\{q_1, q_2, \ldots, q_{n+1}\}$ is said to form a *cycle* if $q_{i+1} = \delta(q_i, \sigma_i), i = 1, 2, \ldots, n$ and $q_1 = \delta(q_{n+1}, \sigma_{n+1})$.

Definition 7 Assume that a set of states $\{z_1, z_2, \ldots, z_{n+1}\}$ in the diagnoser that is F_i-uncertain forms a cycle. Then the set of states $\{z_1, z_2, \ldots, z_{n+1}\}$ is said to form an F_i *-indeterminate cycle* if there exist cycles $\left\{q_1^N, q_2^N, \ldots, q_{k+1}^N\right\}$ and $\left\{q_1^{F_I}, q_2^{F_I}, \ldots, q_{l+1}^{F_I}\right\}$ that satisfies the following.

1. $\left\{q_1^N, q_2^N, \ldots, q_{k+1}^N\right\}$ forms a cycle such that $\kappa(q_m^N) = N, q_m^N \in z_r, m = 1, 2, \ldots, k+1, r = 1, 2, \ldots, n+1, k \leq n$ and $\left\{q_1^N, q_2^N, \ldots, q_{k+1}^N\right\} \subseteq \{z_1, z_2, \ldots, z_{n+1}\}$. This cycle said to be N *-cycle*; and
2. $\left\{q_1^{F_i}, q_2^{F_i}, \ldots, q_{l+1}^{F_i}\right\}$ forms a cycle such that $\kappa(q_{m'}^{F_i}) = F_i, q_{m'}^{F_i} \in z_r, m' = 1, 2, \ldots, l+1, r = 1, 2, \ldots, n+1, l \leq n$ and $\left\{q_1^{F_i}, q_2^{F_i}, \ldots, q_{l+1}^{F_i}\right\} \subseteq \{z_1, z_2, \ldots, z_{n+1}\}$. This cycle said to be F_i *-cycle*.

Theorem 1 *The diagnoser is F_i-diagnosable if and only if there is no F_i-indeterminate cycle in the diagnoser for the failure mode F_i.*

Proof After the occurrence of the failure mode F_i, the state of diagnoser will be one of the following three types: normal, F_i-uncertain and F_i-certain states.

1. In case of normal state condition.
(Sufficiency) Because there is no F_i-indeterminate cycle, there exists a path (sequence or string) to a state whose state condition is F_i. Therefore, the state of diagnoser eventually becomes F_i-certain, which implies that the diagnoser is F_i-diagnosable.
(Necessity) Because the diagnoser is F_i-diagnosable, there exist the path to the state that is F_i-certain in diagnoser. So there is no F_i-inderterminate cycle formed by F_i-uncertain states.
2. In case of F_i-uncertain state condition.
Sufficiency and Necessity are satisfied by the same reasoning as in the case of normal state condition.
3. In the case of F_i-certain state condition.

Because the present state of the diagnoser is F_i-certain, the diagnoser is F_i-diagnosable. Since there is no F_i-indeterminate cycle, there exist a path to the state that is F_i-certain. □

Example 3.2 (*Pump-valve system—F_i-indeterminate cycle*) From Fig. 9, we can observe that the set of states $\{\{q_0, q_7, q_{11}\}, \{q_1, q_8, q_{121}\}, \{q_2, q_{13}\}, \{q_3, q_{10}, q_{14}\}\}$ forms an F_2-inderterminate cycle shown in dash-point box. The reason is that the set of states $\{q_0, q_1, q_2, q_3\}$ forms a cycle with the state condition N while $\{q_{11}, q_{12}, q_{13}, q_{14}\}$ also forms a cycle with the state condition F_2. In other words, there exist both N-cycle and F_2-cycle in DES G as shown in Fig. 4. Therefore, the diagnoser is not F_2-diagnosable because there exists an F_2-indeterminate cycle. The physical meaning of this can be explained by the following. When the failure event Stuck_closed1 or Stuck_closed2 occurs and the control command OV is issued by the supervisor at the initial state $\{q_0, q_7, q_{11}\}$,

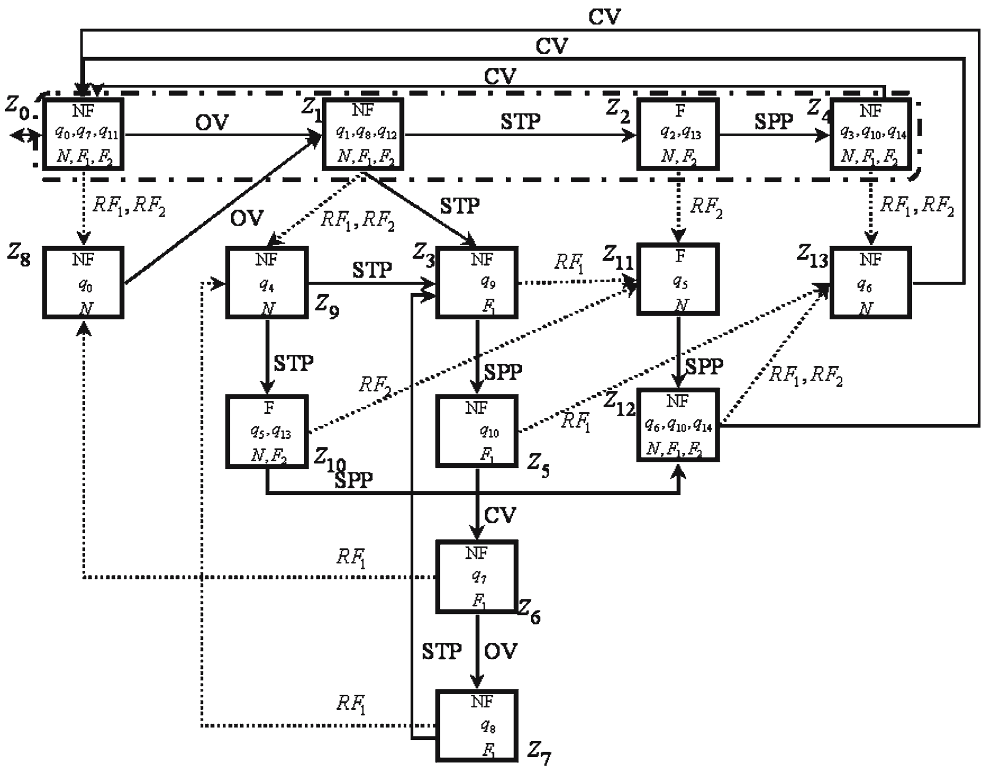

Fig. 9 Recoverable diagnoser of pump-valve system

the state of the diagnoser is F_1-uncertain and F_2-uncertain. This is because the states q_0, q_7, q_{11}, of which state conditions is N, F_1, and F_2, respectively, have the identical sensor output NF. If the value of sensor output remains NF when the control command STP is given, then the state of the diagnoser becomes F_1-certain by reaching state z_3 as shown in Fig. 9. Physically, the diagnoser can detect that the valve is stuck-closed because the value of flowmeter is NF even after the commands sequence OP and STP is given. However, for the case of failure event Stuck_open1 or Stuck_open2, the sensor outputs are exactly the same for both N and F_2 states. This means that the diagnoser cannot escape from the F_2-inderterminate cycle because the diagnoser cannot distinguish N-cycle from F_2-cycle by only looking at the value of flowmeter. Therefore, some additional sensors are needed to detect the failure event Stuck_open1 and Stuck_open2.

Recoverability

In this section we define the recoverability of a recoverable diagnoser and present the necessary and sufficient condition for recoverability. First, the recoverability is defined in Definition followed by the necessary and sufficient conditions for recoverability in Theorem 2.

Definition 8 The recoverable diagnoser is F_i *-recoverable* if

$$\forall\, z_R \text{ such that } \overline{\kappa(z_R)} = \{F_i\}, \quad RF_i \in \gamma_R(z_R)$$

In other words, if the diagnoser can enable the failure recovery event RF_i at all by reaching state z_3 as shown in Fig. 9-certain states, then the diagnoser is said to be recoverable.

Theorem 2 *The recoverable diagnoser is F_i-recoverable if and only if*

Condition 1 *The failure recovery event RF_i can be enabled at any state q_R of which state condition is F_i, and*

Condition 2 *There is no F_i-indeterminate cycle in the recoverable diagnoser.*

Proof (Sufficiency) By the condition 2, the recoverable diagnoser satisfies Theorem 1 so the recoverable diagnoser is F_i-diagnosable. Therefore, there exists a state z_R in the recoverable diagnoser that is F_i-certain. And by the condition 1, at any state z_R that is F_i-certain, it is true that $RF_i \in \gamma_R(z_R)$. Therefore the recoverable diagnoser satisfies the Definition 8 and is F_i-recoverable.

(Necessity) Because the recoverable diagnoser is F_i-recoverable it is true that $RF_i \in \gamma_R(z_R)$ at all state

z_R that is F_i-certain by the Definition 8. Therefore, the failure recovery event RF_i can be enabled at all state $q_R(q_R \in z_R)$ of the DES, and also because z_R is F_i-certain, $\kappa_R(q_R) = F_i$. Then, because the recoverable diagnoser is also F_i-recoverable, there exists the state z_R that is F_i-certain. Therefore, there is no F_i-indeterminate cycle in the recoverable diagnoser. □

Example 3.3 (*Pump-valve system—recoverability*) The recoverable diagnoser shown in Fig. 9 is F_1-recoverable by Theorem 2. The reason is that states z_3, z_5, z_6, and z_7 are F_1-certain and the failure recovery event RF_1 is enabled at those states and that there is no F_1-indeterminate cycle. In physical sense, the recoverability of pump-valve system can be explained by the following. After the control commands sequence OV and STP is issued at the initial state of diagnoser, the diagnoser can detect the failure event Stuck_closed1 or Stuck_closed2 at the state z_3 because the value of sensor output is NF, which is different from the normal cycle. Because RF_1 can be enabled at that state, F_1 can be recovered by issuing RF_1. After RF_1 is issued, the state condition of the system will return to normal.

High-level diagnoser design

Even though the recoverable diagnoser developed in the "Diagnosability and recoverability" section can identify the cause of failure and recover from it, it may be very difficult to design such a diagnoser for large-scale systems commonly found in industrial applications. In order to reduce the size of the diagnoser, this section applies the theory of hierarchical control to the diagnosis problem. We present the model reduction scheme of DES to make high-level diagnoser along with the proof of the equivalency of the original recoverable diagnoser and the high-level diagnoser.

Model reduction

The basic idea of model reduction scheme is to partition states of DES as equivalence classes of sensor output map, control command map, and state condition map because the diagnoser transits and estimates the state by only sensor output, control command, and state condition. So, those states in an equivalence class can be treated identically for the purpose of supervisory control and failure diagnosis.

First, we define the high-level DES.

$$\begin{tikzcd}
Q \arrow[r, "\delta"] \arrow[d, dashed, "P_\pi"'] & Q \arrow[d, "P_\pi"] \\
Q_{hi} \arrow[r, dashed, "\delta_{hi}"] & Q_{hi}
\end{tikzcd}$$

Fig. 10 High-level projection of DES

Definition 9 Define the *high-level DES* as the DES that is reconstructed by the coarsest partition π such that[4]

$$\pi \leq \ker\lambda \wedge \ker\gamma \wedge \ker\kappa$$

The partition π is the coarsest partition that preserves the information on partitions $\ker\lambda$, $\ker\gamma$, and $\ker\kappa$. Also, the high-level projection P_π is defined as $P_\pi : Q \to Q_{hi}$ for the partition π that is illustrated in Fig. 10.

The states of a DES are aggregated by high-level projection P_π into a high-level state that has the same sensor output, control command, and state condition information.

Define the high-level DES based on Definition 9 as FSMA

$$G_{hi} = \left\{Q_{hi}, \Sigma_{hi}, \delta_{hi}, q_{hi,0}, Q_{hi,m}, Y_{hi}, \lambda_{hi}, C_{hi}, \gamma_{hi}\right\} \tag{10}$$

where G_{hi}, Σ_{hi}, δ_{hi}, $q_{hi,0}$, and $Q_{hi,m}$ are the state set, the event set, the transition function, the initial state, and the marked state set, respectively; and Y_{hi}, λ_{hi}, C_{hi}, and γ_{hi} are the sensor output set, the sensor output map, the control command set, and the control command map, respectively.

Example 4.1 (*Pump-valve system—high-level model*) We obtain the high-level DES for the DES of the pump-valve system in Example 2.1 as shown in Fig. 4 using the projection P_π with $\pi \leq \ker\lambda \wedge \ker\gamma \wedge \ker\kappa$. Partitions $\ker\lambda$, $\ker\gamma$, and $\ker\kappa$ are easily obtained from Tables 1, 2, and 3, respectively. The resulting partitions are shown in (11), (12), and (13). And partition π is $\pi = \ker\lambda \wedge \ker\gamma \wedge \ker\kappa = \ker\gamma$. Three states are reduced from the original DES. So, the number of high-level state is 12. Let us rename the states of high-level DES as $Q_{hi} = \left\{q_{hi,0}, q_{hi,1}, \ldots, q_{hi,11}\right\}$. The high-level DES is shown in Fig. 11.

$$\ker\lambda = \{q_0, q_1, q_3, q_4, q_6, q_7, \ldots, q_{12}, q_{14}\} \dot{\cup} \{q_2, q_5, q_{13}\} \tag{11}$$

[4] Notation $\ker\pi$ means the coset (equivalent class) for the binary relation π.

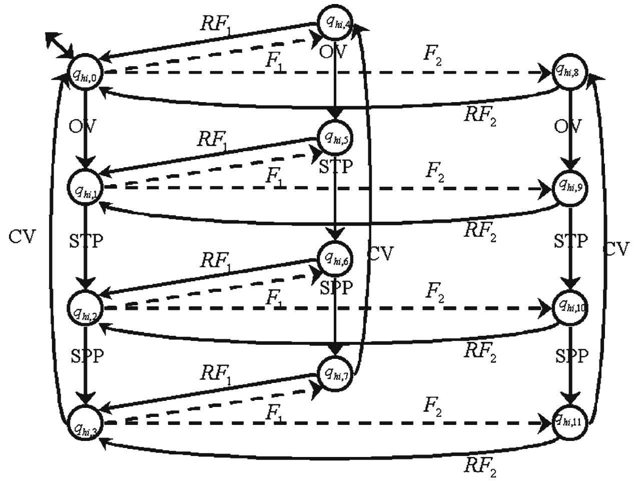

Fig. 11 High-level DES of pump-valve system

$$\ker\gamma = \{q_0\} \dot\cup \{q_1, q_4\} \dot\cup \{q_2, q_5\} \dot\cup \{q_3, q_6\} \dot\cup \{q_7\} \dot\cup \{q_8\} \dot\cup \ldots \dot\cup \{q_{14}\} \tag{12}$$

$$\ker\kappa = \{q_0, q_1, \ldots, q_6\} \dot\cup \{q_7, \ldots, q_{10}\} \dot\cup \{q_{11}, \ldots, q_{14}\} \tag{13}$$

The pairs of states (q_1, q_4), (q_2, q_5), and (q_3, q_6) form the cosets by the partition π. We can confirm this by the FSMA G in Fig. 4, the sensor output map in Table 1, the control command map γ in Table 2, and state condition map in Table 3. For example, states q_1 and q_4 have the same sensor output, NF; the same control command, OV; and the same state condition, N.

High-level diagnoser

Let the FSMA

$$D_{hi} = \left\{Z_{hi}, E_{hi}, \varsigma_{hi}, z_{hi,0}, Z_{hi,m}, \overline{K_{hi}}, \overline{\kappa_{hi}}\right\} \tag{14}$$

denote the high-level diagnoser for the high-level DES. For convenience, we refer to the high-level recoverable diagnoser as the high-level diagnoser hereinafter. In (14), Z_{hi}, E_{hi}, ς_{hi}, $z_{hi,o}$, and $Z_{hi,m}$ are the state set, the event set, the transition function, the initial state and the marked state set, respectively. And $\overline{K_{hi}}$ and $\overline{\kappa_{hi}}$ are the state condition set and state condition map, respectively.

We can also define the diagnosability and recoverability for high-level diagnoser as given on Definition 5 and 8, respectively. And the necessary and sufficient condition for diagnosability and recoverability are also the same as proven in Theorem 1 and 2.

Example 4.2 (*Pump-valve system—high-level diagnoser*) By the similar way to Example 3.2, high-level diagnoser can be designed from high-level DES shown in Fig. 11. Figure 12 shows the high-level diagnoser for the pump-valve system where two states are reduced from the original diagnoser.

Two states are reduced from the original diagnoser as shown in Fig. 9.

Equivalency

We show that the high-level diagnoser is equivalent to the original recoverable diagnoser in the following .

Theorem 3 *The recoverable diagnoser D and the high-level diagnoser D_{hi} are equivalent.*

Proof We have to show that the state condition of D and D_{hi} are identical for the state pair (q, q_{hi}) that has the sensor output and control command. That is, we have to show the following equation holds.

$$\left[\forall\lambda(q) = \lambda_{hi}(q_{hi}) \ \wedge \ \gamma(q) = \gamma_{hi}(q_{hi})\right], \ \overline{\kappa(z)} \ = \ \overline{\kappa_{hi}(z_{hi})}$$

First, because $P_\pi(q_k) = q_{hi,k}$ and $\pi \ \leq \ \ker\lambda \wedge \ker\gamma$, for z_k such that $q_k \in z_k$ and $z_{hi,k}$ $\overline{\text{such that}}$ $q_{hi,k} \in z_{hi,k}$, it follows that $P_\pi(z_k) = z_{hi,k}$ and $\overline{\kappa_{hi}(z_{hi,k})} \ = \ \overline{\kappa_{hi}\{P_\pi(z_k)\}}$. $\overline{\text{Then}}$, from $\pi \ \leq \ \overline{\ker\kappa}$ it follows that $\overline{\kappa_{hi}\{P_\pi(z_k)\}} \ = \ \overline{\kappa(z_k)}$. Therefore, $\overline{\kappa(z_{hi,k})} \ = \ \overline{\kappa(z_k)}$. □

Example 4.3 (*Pump-valve system—equivalency*) Even though the number of states of high-level diagnogser is fewer than that of the original diagnoser, it produces the identical estimation with the recoverable diagnoser for the pump-valve system because the high-level diagnoser is also F_1-diagnosable and F_1-recoverable.

Conclusion

This paper presents a new approach for on-line passive diagnoser that is capable of not only the failure diagnosis but also the supervisory control. This new approach is a hybrid of event-based and state-based strategies along with the introduction of failure recovery events and hierarchical control concept. The contributions of the paper can be summarized as follows.

This paper establishes a new failure diagnosis approach based on the combination of two well-known approaches, i.e., event-based and state-based. Compared to the event-based approach, the new approach can construct a simpler diagnoser because there is no need to update the state condition for all observable events. In comparison with the state-based approach, the hybrid approach allows much simpler implementation because there is no need to synchronize the diagnoser with the supervisor of the system.

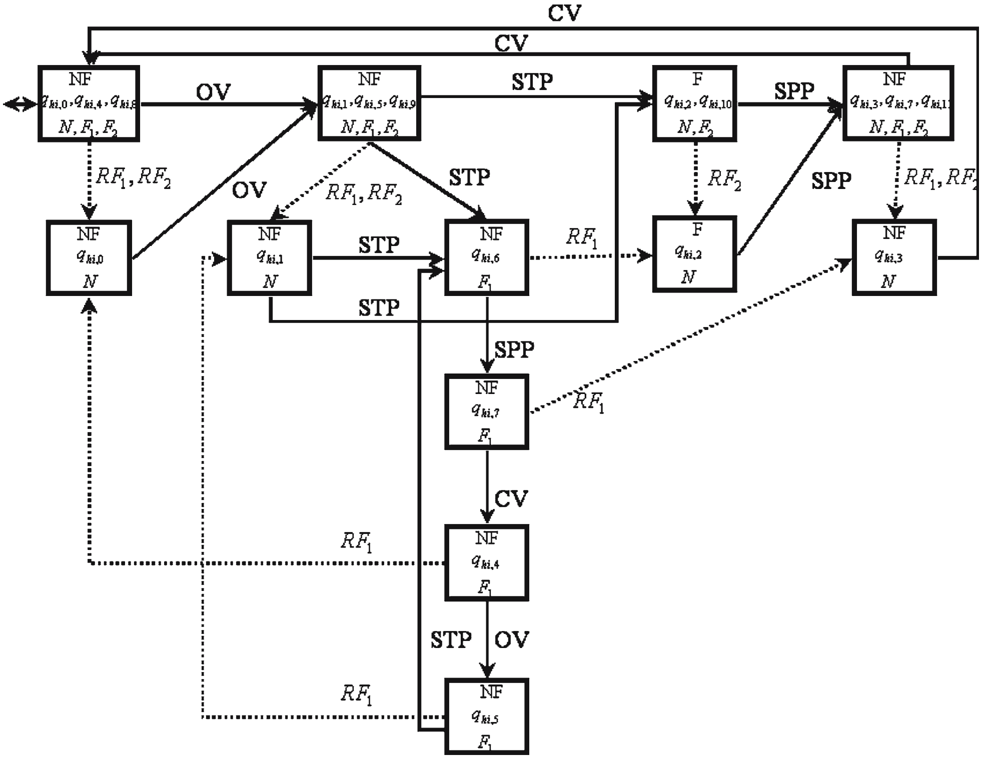

Fig. 12 High-level diagnoser of pump-valve system

By introducing failure recovery events, the supervisory controller and failure diagnoser can be integrated so that an action can be taken for failure recovery once the failure is diagnosed.

In order to apply this approach to real-world problems where the system quickly becomes too large to be handled, the theory of hierarchical control is applied to the diagnoser design. The high-level diagnoser performs exactly identical functions of the original diagnoser with less number of states.

References

Brandt, R. D., Garg, V., Kumar, R., Lin, F., Marcus, S. I., & Wonham, W. M. (1990). Formulas for calculating supremal controllable and normal sublanguages. *System Control Letter, 15*(2), 111–117.

Cassandras, C. G. (1993). *Discrete event systems: Modeling and performance analysis*. Richard D. Irwin, Inc.

Davey, B. A., & Priestley, H. A. (1990). *Introduction to lattices and order*. Cambridge University Press.

Hashtrudi Zad, S. (1999). Fault diagnosis in discrete-event and hybrid systems. Ph.D. Thesis, The University of Toronto.

Hashtrudi Zad, S., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Transaction on Automatic Control, 48*(7), 1199–1212.

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages and computation*. Addision-Wesley.

Kumar, R., & Garg, V. (1995). *Modeling and control of logical discrete event systems*. Kluwer Academic Publishers.

Lin, F., & Lin, T. W. (1993). Diagnosability of discrete event systems and its applications to circuit testing. *Proceedings of the 36th Midwest symposium on circuits and systems*.

Park, Y. (1996). Model-based monitoring of discrete event systems. Ph.D. Thesis, The Purdue University.

Ramadge, P. J., & Wonham, W. M. (1989). The control of discrete event systems. *Proceeding of IEEE, 77*, 81–98.

Sampath, M. (1995). A discrete event systems approach to failure diagnosis. Ph.D. Thesis, The University of Michigan.

Sampath, M., Lafortune, S., & Teneketzis, D. (1998). Active diagnosis of discrete-event systems. *IEEE Transaction on Automatic Control, 43*(7), 908–929.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transaction on Automatic Control, 40*(9), 1555–1575.

Wonham, W. M. (1998). *Notes on control of discrete event systems*. Department of Electrical and Computer Engineering, University of Toronto.

Wonham, W. M., & Ramadge, P. J. (1987). On the supremal controllable sublanguage of a given language. *SIAM Journal of Control and Optimization, 25*(3), 637–659.