Asian Journal of Control, Vol. 12, No. 3, pp. 309–322, May 2010
Published online 1 April 2010 in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/asjc.187

CONSTRUCTING AN ABSTRACT MODEL FOR LADDER DIAGRAM DIAGNOSIS USING PETRI NETS

Jui-I Tsai and Ching-Cheng Teng

ABSTRACT

This paper proposes a method for constructing an abstract model for analyzing and diagnosing electrical circuit ladder diagrams (LDs) using Petri nets, and also supporting network-based monitoring and supervision. This approach converts normal open (NO) and normal close (NC) contacts in the LD into Petri net transitions, and converts devices (*e.g.* relay coils) in the LD into Petri net places. This study introduces the concepts of composite transitions, composite places, and relevant state to reduce complexity and increase readability of Petri nets for constructing abstract models. The current study constructing diagnosis of fault modeling, introduces simple matrix manipulation and the difference output vector (DOV) to determine the faulty area for diagnosis in the ladder diagram. An LD controller example demonstrates the usable approach.

Key Words: Ladder diagram; Petri net; diagnosis; abstract model; BPN; fault model.

I. INTRODUCTION

Researchers developed ladder diagrams (LDs) to replace relay symbols, and these diagrams have been popular for a long time. Using a LD, it is possible to represent the automated control process both sequentially and graphically.

Petri nets (PNs) describe and analyze information flow, and are excellent tools for modeling asynchronous concurrent systems, such as computer systems, manufacturing systems, Supply Chain Management [1], video streaming systems [2] and power systems [3, 4].

Sequence controllers play a key role in industrial practice or manufacturing systems. Traditionally, ladder diagrams have been widely applied to programmable logic controllers, while Petri nets have supplied an alternative tool for the sequence control of complex systems.

Jackman *et al.* [5] proposed a conceptual model and working equation for converting relay ladder logic to the PN model. Lee *et al.* [6] presented a method for obtaining an augmented PN from an LD, and then using the Petri net state equation as an analysis technique to validate the corresponding flow mechanism of the generated PNs. However, this approach increases the total number of nodes and links in the generated PNs [7]. Venkatesh *et al.* [8, 9] and Peng *et al.* [7] modeled an LD contact to place and increase a virtual transition. Lee *et al.* [10] modeled an LD connect to transition and increase a position. However, the total number of nodes and links in the generated Petri nets increases, which in turn increases complexity. Petri nets have an excellent flexibility corresponding to different models according to a variety plant. This paper proposes a Boolean Petri Net (BPN) Model.

Hierarchical control is an approach to designing large-scale discrete event systems to deduce complexity [10] as Fig. 1 illustrates. In a manufacturing system, an LD controller may use a local controller, enabling the LD controller to be diagnosed and monitored

Manuscript received February 15, 2009; revised June 5, 2009; accepted November 13, 2009.

The authors are with the Department of Electrical Engineering, National Chiao-Tung University, Hsinchu, Taiwan (e-mail: tsai.ece90g@nctu.edu.tw, ccteng@cn.nctu.edu.tw).

The authors thank the Guest Editors and reviewers for their thoughtful and constructive comments that helped greatly improve the presentation of this paper.

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

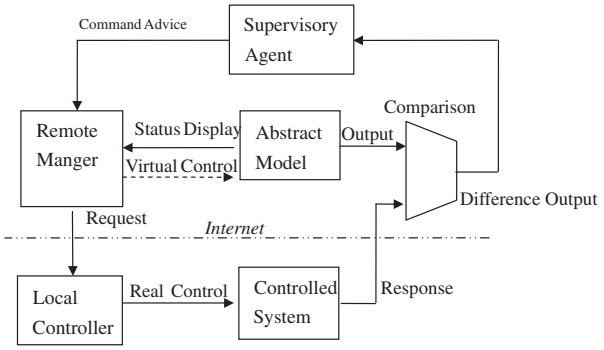


Fig. 1. Proposed hierarchical control [10].

remotely. This paper models the local controller (*i.e.* LD controller) and abstract model (*i.e.* corresponding to the LD model) using a BPN. The LD controller model is a structural model, and the abstract model is a behavioral model. The structural model is similar to the original LD architecture, while the behavioral model is simplified through the structural model but still matches the functions of the LD controller.

The paper is organized as follows. Section II defines the BPN model, and describes the properties of a PN. Section III builds a table of a one or more rung LD corresponding to the BPN module using the Boolean equation. Section IV presents an example of transferring the LD to the abstract model, discusses the properties of the proposed abstract model, and simulates fault free and fault models of the proposed BPN. Finally, Section V gives the conclusion.

II. BOOLEAN PETRI NET

Carl Adam Petri proposed the Petri net theory. Fig. 2 shows the structure of Petri nets in a directed bipartite graph that consists of places, transitions, and arcs. A circle with a token represents the places. A bar that indicates the flow of tokens when firing condition is satisfied, which represents the transition. Finally, a straight line that connects the place to the transition, or the transition to the place denotes the arc, which indicates the flow of tokens in the direction of the arrow.

2.1 Definition of boolean petri nets

The purpose of developing the BPN model is that this model exhibits the imply logic property in a LD. The simplest way to represent an LD is by its Boolean equation. The approach proposed in this paper embeds the Boolean equation in a PN transition to develop the BPN model. This special transition is called the "Boolean transition." Table I describes the BPN model

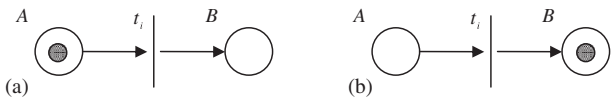


Fig. 2. (a) An example Petri net, (b) A token moving from A to B in Fig. 2a after t_i fire.

corresponding to the Boolean equation. Clearly, the BPN model also matches the LD. To map an LD into a Petri net, the Petri net must be extended. This extended Petri net is called a Boolean Petri net, which can be defined formally as

$$PN = (P, T, A, I, O, in, out, M_0) \tag{1}$$

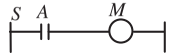
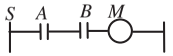
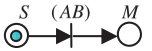
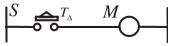
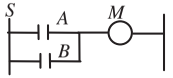
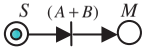
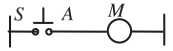
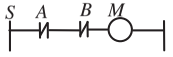
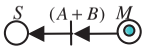
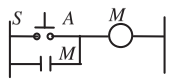
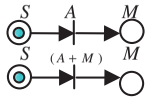
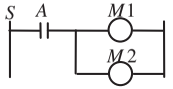
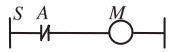
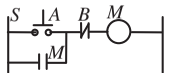
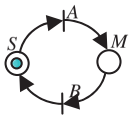
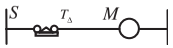
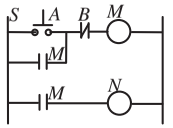
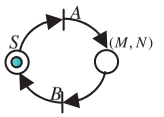
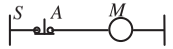
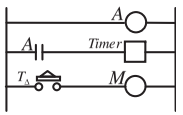
where $P = \{p_1, p_2, \ldots, p_m\}$, $m \geq 1$, is a finite set of places representing the LD action state. The places are associated with a component or a set of components (*i.e.*, a compound component) such as the actuator output, relay coil, timer, counter solenoid, or source; $T = \{t_1, t_2, \ldots, t_n\}$, $n \geq 1$, is a finite set of transitions representing event whether occurs or not. These transitions are always associates with a switch or a set of switches and represented by Boolean equations or variables.

The switch can be a normal open (NO) switch or normal closed (NC) switch. The NO switch is also called an "*a*" contact and the NC switch is also called a "*b*" contact, where $P \cap T = \Phi$ and $P \cup T \neq \Phi$.

$A \subset (P \times T) \cup (T \times P)$ is a set of arcs $(\rightarrow)$ consisting of input arcs $A_i(P \times T)$ and output arcs $A_o(T \times P)$. The weight of each directed arc in this paper is 1, and $A_i(P \times T)$ is defined as directed arcs from a place to a transition. Places are called input places and transitions are called output transitions, and the input arc is represented by a connected line as channel of token. $A_o(T \times P)$ is defined as directed arcs from a transition to a place, the transition is called the input transition and the place is called the output place, and the output arc is represented by a connected line as channel of token. The arc may be preservation arc $(\bullet\rightarrow)$ that a input arc and a output arc exist simultaneously between same place and transition [6]; $I: T \times P \rightarrow N$ is an input function that defines as number of output arcs $A_o(T \times P)$, where $N = \{0, 1, 2, \ldots\}$, $O: P \times T \rightarrow N$ is an output function that are defined as number of input arcs $A_i(P \times T)$, where $N = \{0, 1, 2, \ldots\}$; $in = \{in_1, in_2, \ldots, in_n\}$ is a set of input switch, which is represented by Boolean function or variable. A set of input switches is associated with a transition t_j and is denoted by $t_j = \{in\}$. The Boolean function or variable can be '1', in which case the related transition t_j is allowed to fire if it is enabled, or it can be '0', in which case the related transition t is

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

Table I. Some LD Modules and their Corresponding BPN Models.

Modules	Ladder Diagrams	Boolean Equations	Boolean Petri Nets	Modules	Ladder Diagrams	Boolean Equations	Boolean Petri Nets
Type 1		$M = A$		Type 3		$M = AB$	
		$M = T_\Delta$		Type 4		$M = A + B$	
		$M = A$		Type 5		$M = \overline{AB}$ $\overline{M} = A + B$	
		$M = A + M$ $M = A$		Type 6		$M1 = A$ $M2 = A$ $M1 = M2$	
Type 2		$\overline{M} = A$		Type 7		$M = (A + M)\overline{B}$ $= A\overline{B}$	
		$\overline{M} = T_\Delta$		Type 8		$M = (A + M)\overline{B} = A\overline{B}$ $N = M$	
		$\overline{M} = A$		Type 9		$Timer = A$ $M = T_\Delta$	

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

Table I. *Continued.*

Modules	Ladder Diagrams	Boolean Equations	Boolean Petri Nets	Modules	Ladder Diagrams	Boolean Equations	Boolean Petri Nets
Type 10		$\overline{M}=A$ $N=A$		Type 11		$\overline{B}=T_\Delta$ $C=T_\Delta$ $M=A\overline{N}$ $N=B\overline{M}$ $\overline{M}=A\overline{\overline{N}}=\overline{A}+N$ $=\overline{A}+B\overline{M}$ $(1+B)\overline{M}=\overline{A}$ $\overline{M}=\overline{A}$ $\Rightarrow M=A$	
Type 12		$\overline{B}=T_\Delta+C$ $C=T_\Delta\overline{B}$ $\overline{B}=T_\Delta+C$ $=T_\Delta+T_\Delta\overline{B}$ $(1+T_\Delta)\overline{B}=T_\Delta$ $\Rightarrow\overline{B}=T_\Delta$ $C=T_\Delta\overline{B}$ $=T_\Delta(T_\Delta+C)$ $(1+T_\Delta)C=T_\Delta$ $\Rightarrow C=T_\Delta$		Type 13		$\overline{N}=B\overline{M}=\overline{B}+M$ $=\overline{B}+A\overline{N}$ $(1+A)\overline{N}=\overline{B}$ $\overline{N}=\overline{B}$ $\Rightarrow N=B$	
Type 14		$Timer=A$ $B=A(\overline{T_\Delta})$ $C=T_\Delta$					

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

not allowed to fire; $out=\{out_1, out_2, \ldots, out_m\}$ is a set of output actuator which is associated with a p_i and is denoted by $p_i=\{out\}$; $M_0(P)$ is the initial marking that uses a token to represent the place status.

A transition is enabled if the number of tokens at the place is larger than or equal to the number of input arcs. A transition is firing if the enabled transition is fired and its transition states are true (*i.e.*, the Boolean equation is true). When a transition fires, it moves the tokens from input places to output places along the input arcs and output arcs, as Fig. 2 illustrates. This moves the token of place A to place B along directed arcs if transition t_i is firing. A marking is denoted as an m-vector, where m is the total number of place P, while $m(p_i)$ is represented by the number of tokens at place p_i [11].

For the marking m_0, there is an enabled transition t_1. If there is a firing of transition t_1, then the marking is immediately reachable to m' from m_0, denoted by $m_0[t_1>m'$. A marking m_i is said reachable from m_0 if there exists a sequence of firings that transforms m_0 to m_i. $R(m_0)$ is defined as the set of all reachable markings from m_0. $F(m_0)$ is defined as the set of all firing sequences from m_0. A place p_i is said to be bounded for an initial marking m_0 if $\exists k>0$, $m(p_i)\le k$, and $\forall m\in R(m_0)$. Specifically, it is said to be safe if $k=1$. A marking m_0 is said to be live for a Petri net if every marking has been reached from m_0, which indicates it is possible to fire any transition of the net by some firing sequence [12, 13]. If m_0 may be reached from any marking, The Petri net is said to be reversible.

To simulate the behavior of LD, this approach changes a state or marking according to defined firing rules for the Boolean Petri nets model.

2.2 State equation

The firing definition easily shows that the token moves from state M_{k-1} to another state M_k by the kth firing, and U_k is a firing vector which can be given in terms of the following matrix state equation for Petri nets [12]

$$M_k = M_{k-1} + A^T U_k \tag{2}$$

where U_k is called firing vector, and A^T is called the incidence matrix for any given topological structure of Petri nets, defined by

$$A^T(p_i,t_j)=\begin{Bmatrix} -A_o(p_i,t_j) \\ 0 \\ A_i(t_j,p_i) \end{Bmatrix}, \text{ where } 1 \le i \le m, \quad 1 \le j \le n. \tag{3}$$

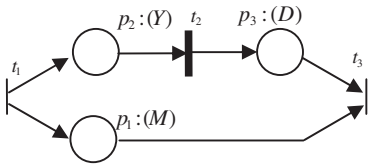


Fig. 3. A simple example.

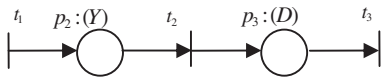


Fig. 4. The reduction result of Fig. 3.

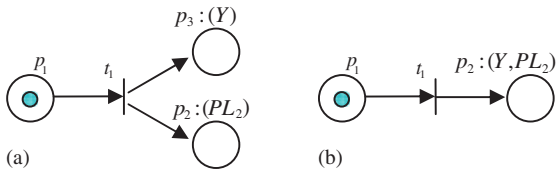


Fig. 5. (a) A simple example and (b) The composite result of Fig. 5a.

Note that M_k must be a vector of nonnegative integers [12]. The firing vector will then select an appropriate column of A^T such that

$$M_{k-1} + A^T U_k \ge 0 \quad \text{for each } k \tag{4}$$

2.3 Definition of action dominance and equivalence

Dominance. An action p_1 is said to dominate another action p_2 in an irredundant place iff every exist of token for p_2 is also exist of token for p_1. *i.e.*, the life of a token of p_1 is longer than p_2, denoted as $m(p_1)>m(p_2)$. The reduction of the place p_1 to be analyzed is based on the dominance relation.

Example 1. Fig. 3 shows a PN in which p_1 is dominated by p_2 and p_3, *i.e.*, $m(p_1)>m(p_2)$ and $m(p_1)>m(p_3)$. Fig. 4 shows the reduction result.

Equivalence. The actions p_1 and p_2 are equivalent if exist of token is same condition for p_1 and p_2, *i.e.*, $m(p_1)>m(p_2)$ and $m(p_2)>m(p_1)$. The composite of the place p_1 and p_2 to be analyzed is based on the equivalent action.

Example 2. Fig. 5a shows a PN in which p_2 is equivalent to p_3, *i.e.*, $m(p_3)>m(p_2)$ and $m(p_2)>m(p_3)$. Fig. 5(b) shows the composite result.

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

III. LADDER DIAGRAM MODEL USING BOOLEAN PETRI NET

3.1 Model of basic modules

In ladder diagrams, the horizontal line (rung) and the associated elements represent Boolean equations. Similarly, in Boolean Petri nets, the associated transitions represent Boolean equations. In ladder diagrams, the symbol "◯" represents the dependent element of the equation (coil). Similarly, in Boolean Petri nets, the symbol "◯" represents the dependent element of the equation (place). In ladder diagrams, "| |" represents the independent element (normal open contacts), while in Boolean Petri nets, "| or |" represents the independent element (input transitions). A diagonal line placed in the middle of these symbols (*i.e.*, "|/|") represents normal closed contacts, which indicate that the negated value of the variable is used. Similarly, bar "| or |" represents the output transition. In ladder diagrams, variables (contacts) placed in a series represent the *AND* Boolean function, while contacts placed in parallel represent the *OR* Boolean function. The rungs are executed in order from top to bottom. Therefore, the Type 8 ladder diagram in Table 1 represents Boolean equations $M=(A+M)\overline{B}$ and $N=M$ [14]. In Boolean Petri nets, a similar input transition represents $(A+M)$, denoted as $t_1:(A+M)$, which is a composite transition. Conversely, output transitions represent $\overline{B}$, denoted as $t_2:(B)$, and output places are denoted as $p_2:(M,N)$ which are composite places. Finally, Table I summarizes some typical LD modules and their corresponding Boolean Petri net models, where S is a pseudo source and the composite and decomposite of Boolean Petri net are as shown in Table II.

3.2 Model of faulty ladder diagram

An LD circuit fault may generally be classed as both stuck-at 0 (s-a-0) and stuck-at 1 (s-a-1) type; the stuck-at 0 fault is like a NO switch and the stuck-at 1 fault is like an NC switch. Hence the fault model of the ladder diagram can be modeled as a Petri net. An LD of the possessed example fault is illustrated in Fig. 6a and the Petri net model is illustrated in Fig. 6b. Where fault f_1 are represented s-a-0 to represent the switch A is struck at open, fault f_2 is represented s-a-1 to represent the switch B is stuck at close. According to Eq. (3), the incidence matrix is

$$A^T = \begin{matrix} \\ p_1 \\ p_2 \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \end{matrix}.$$

The faults classified in the two cases are interpreted as below.

Case 1. Assume f_1 is s-a-0 fault and initial marking is

$$M_0 = \begin{matrix} p_1 \\ p_2 \end{matrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

When the bottom is pushed $A=1$.

$$U_1 = \begin{matrix} t_1:(A) \\ t_2 \end{matrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad U_{f_1} = \begin{matrix} t_1:(A\cdot f_1) \\ t_2 \end{matrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$U_{1/f_1} = \begin{matrix} t_1:(A/A\cdot f_1) \\ t_2 \end{matrix} \begin{bmatrix} 1/0 \\ 0 \end{bmatrix},$$

where U_1 and U_{f_1} are represented as the fault free and faulty firing vector, respectively. $U_{1/f}$ is represented as the fault free/faulty firing vector.

According to Eq. (2)

$$M_{1/f_1} = M_0 + A^T U_{1t/f} = \begin{matrix} p_1 \\ p_2 \end{matrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{matrix} \\ p_1 \\ p_2 \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} 1/0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1/0 \\ 1/0 \end{bmatrix} = \begin{bmatrix} 0/1 \\ 1/0 \end{bmatrix},$$

i.e., $M_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; $M_{f_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, where M_1 and M_{f_1} are represented as the fault free and faulty marking vector, respectively. $M_{1/f}$ is represented as the fault free/faulty making vector.

In the LD circuit, f_1 fault means the coil C is not active since the switch A is stuck at open.

Case 2. Assume f_2 is s-a-1 fault and current marking is

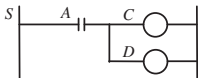
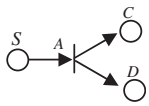
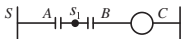
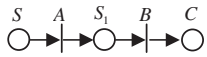
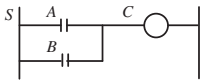
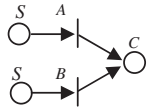
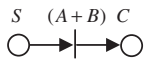
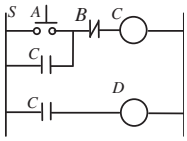
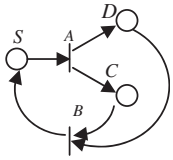
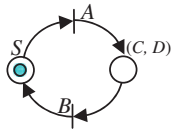
$$M_1 = \begin{matrix} p_1 \\ p_2 \end{matrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

When the bottom is pushed $B=1$

$$U_2 = \begin{matrix} t_1 \\ t_2:(B/B\cdot \overline{f}_2) \end{matrix} \begin{bmatrix} 0 \\ 1/0 \end{bmatrix}$$

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

Table II. Composite and Decomposite of Boolean Petri nets.

LD	Boolean Equation	BPN Decomposite	BPN Composite
	$C = A$ $D = A$ $C = D$		
	$C = AB$		
	$C = A$ $C = B$ $C = A + B$		
	$C = A\overline{B}$ $D = C$		

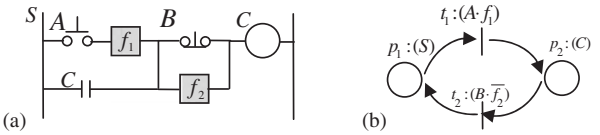


Fig. 6. (a) An LD of possessed fault and (b) A Petri net model of possessed fault.

$$M_{2/f_2} = M_1 + A^T U_2 = \begin{matrix} p_1 \\ p_2 \end{matrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$+ \begin{matrix} \\ p_1 \\ p_2 \end{matrix} \begin{matrix} t_1 & t_2 \\ \end{matrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1/0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 1/0 \\ -1/0 \end{bmatrix} = \begin{bmatrix} 1/0 \\ 0/1 \end{bmatrix}$$

i.e., $M_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$; $M_{f_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

In the LD circuit, f_2 fault means the coil C is maintain action since the switch B is stuck at close.

IV. APPLICATION EXAMPLE

This section illustrates a practical example of hierarchical control system in Fig. 1. The local controller is an LD circuit. This circuit can be modeled by BPN and is simplified to obtain an abstract model using Table I of the preceding section. The system fault can be diagnosed by the difference between the LD response and abstract model response. The differences are as decision of supervisor agent.

4.1 Constructing an abstract model using a Boolean PN

To start a three-phase motor, an LD controller use type of Y-Δ starting to limit starting current, as shown in Fig. 7 and symbol descriptions in Table III. In the LD controller, the bottom Pb_1 is control relay coil M, Y and timer coil active. The motor enters the starting state when NO contacts of M and Y are turned on. Next, the relay coil Y turns off after delay time T_Δ, and the motor returns to the normal state when the relay coil Y turns off and relay coil D turns on. Finally, the motor stops if the bottom Pb_2 is pushed or the current is overload. This LD controller can be specified as follows:

Step (1) The motor is commanded to start (Pb_1).
Step (2) The motor starting time is T_Δ.
Step (3) The motor is commanded to stop (Pb_2).
Step (4) The motor will stop if the current is overloaded.

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

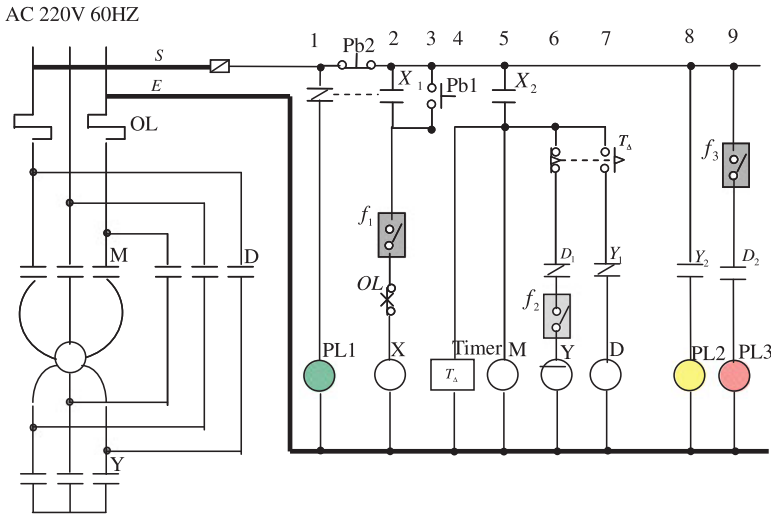


Fig. 7. Control circuit of a Y-Δ starting motor.

Table III. The Descriptions of Symbol.

Symbol	Description	Symbol	Description
PL1	Indicator light of green		"*b*" contact of Push bottom
PL2	Indicator light of yellow		"*a*" contact of Push bottom
PL3	Indicator light of red		"*a*" contact of relay
	Relay		"*b*" contact of relay
T_Δ	Timer		"*a*" contact of timer
	Stuck at 0 (s-a-0) switch for simulate fault [15]		"*b*" contact of timer
1∼9	rung number		"*b*" contact of over load

The implicit specification is as following:

Spec) The relay coil *D* and relay coil *Y* are mutually exclusive.

The transformation from the ladder diagram (in parallel) to the abstract model (in series) is based on the following steps:

Step (1) A rung or compound rung of LD is converted to a Boolean Petri net module using Table I or the Boolean equation. LD controller then assembles Boolean Petri net modules, as Fig. 8 shows, where (1), (2) ... and (9) correspond to the number of LD rungs.

Step (2) A Boolean Petri net can be given after eliminating the redundant or pseudo places (*i.e.*, the *S* place), as Fig. 9 illustrates.

Step (3) An abstract model can be obtained according to dominance relation reduce some places (in this case, an abstract model reduce place p_2), and eliminating some redundant elements (*i.e.*, the coil of time or auxiliary relay), as Fig. 10 shows.

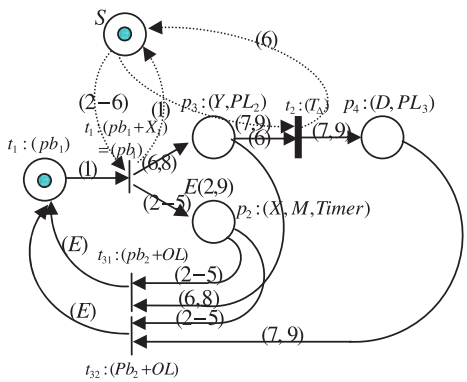


Fig. 8. BPN model of an LD controller.

4.2 Properties of the proposed Boolean Petri net

The reachability of a PN is a tree, which uses states as nodes and transitions as arcs [16]. The construction of this tree starts from the root node. The root node is represented as the initial state, and the arcs outgoing from the root node are marked by the corresponding enabled

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

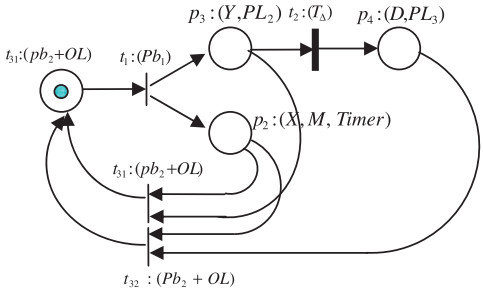


Fig. 9. Equivalent diagram of Fig. 8.

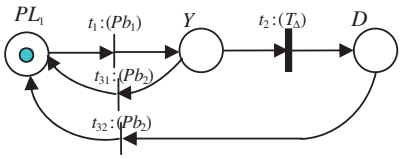


Fig. 10. Abstract model of Fig. 9.

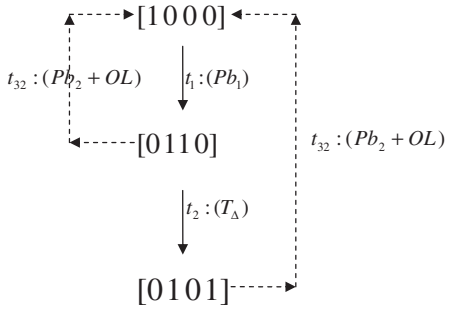


Fig. 11. The reachability tree of the Proposed BPN.

transition. The arc will outgo to a new node (state) from the firing of the corresponding transition (arc). The above procedures are repeated until they produce duplicate nodes. Terminal nodes are identical to existing nodes, which have no any enable transitions are met. In the reachability tree, dash lines indicate nodes.

Due to the similar processes of PN reachability tree, this study presents the reachability tree of the proposed BPN in Fig. 9. For the sake of simplicity in representing the node (node) in the reachablility tree, define the state variable vector in the reachability tree as [p1 p2 p3 p4], and allow the initial state to be [1 0 0 0]. If transition $t_1:(Pb_1)$ fires when $Pb_1=1$ (*i.e.* Pb_1 is active), the state moves to [0 1 1 0]. If transition $t_{31}:(Pb_2)$ fires when $Pb_2=1$ (*i.e.* Pb_2 is active), then the state moves to [1 0 0 0]. Subsequently, if transition $t_2:(T_\Delta)$ is enabled and fires, then the state moves to [0 1 0 1], while if transition $t_{32}:(Pb_2)$ is enabled and fires, the state moves to [1 0 0 0].

Proposition 1. The proposed BPN is live.

Proof. Consider a case based on the reachability tree in Fig. 11. This figure shows that there is no terminal node. Therefore, there always exist some sample path such that any transitions can eventually fire to reach any states from the initial state p_1, *i.e.* $R(m_0)=\{p_2, p_3, p_4\}$, $F(m_0)=\{t_1, t_2\}$. According to this definition, the proposed BPN is live. □

Proposition 2. The proposed BPN is reversible.

Proof. The reachability tree in Fig. 11 indicates that there is no terminal node. Therefore, there always exist some sample path such that any transitions can eventually fire to reach the initial state p_1 from any states (*i.e.* p_2, p_3, p_4), $p_1 \in R(p_2)$, $p_1 \in R(p_3)$, $p_1 \in R(p_4)$. According to this definition, the proposed BPN is reversible. □

Proposition 3. The proposed BPN is bound.

Proof. In a stable PN, the number of tokens in any place will not grow infinitely. The reachability tree in Fig. 11 indicates that one and only one marked token corresponds to any specific state. Therefore, the number of marked tokens in p_1, p_2, p_3 and p_4 is bounded above by 1. According to this definition, the proposed BPN is bounded and safe. □

Similarly, the proposed abstract model in Fig. 10 is live, reversible, and safe. In the Petri net model, the live is represented as a reachable starting state (*i.e.* Y state) and running state (*i.e.* D state) from the ideal state (*i.e.* PL_1 state), the safe is represented as only existing in one state, the reversible is represented as returning to the ideal state (*i.e.* PL_1 state) from any other state (*i.e.* Y state and D state).

4.3 State equation

According to Eq. (2), Fig. 9 shows that the state equation is $M_k = M_{k-1} + A^T U_k$

$$A^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_{31} & t_{32} \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{array},$$

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

$$M_0 = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, U_1 = \begin{matrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$U_2 = \begin{matrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, U_{31} = \begin{matrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, U_{32} = \begin{matrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$M_1 = M_0 + A^T U_1 = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{matrix} \\ t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 & t_{31} & t_{32} \end{matrix} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$M_2 = M_1 + A^T U_2 = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$+ \begin{matrix} \\ t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 & t_{31} & t_{32} \end{matrix} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$M_{31} = M_1 + C^T A^T U_{31} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$+ \begin{matrix} \\ t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 & t_{31} & t_{32} \end{matrix} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ -1 \\ 0 \end{bmatrix} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$M_{32} = M_2 + A^T U_{32} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$+ \begin{matrix} \\ t_1 \\ t_2 \\ t_{31} \\ t_{32} \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 & t_{31} & t_{32} \end{matrix} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \\ -1 \end{bmatrix} = \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In reality, places $p_2:(X, M, Timer)$, $p_3:(Y, PL_2)$, and $p_1:(D, PL_3)$ are compounded places in Petri nets. Therefore, the state of $P=\{p_1, p_2, p_3, p_4\}$ can be decompounded into the $P=\{PL_1, X, M, Timer, Y, PL_2, D, PL_3\}$ state, so $M_0^T(P)=[1\ 0\ 0\ 0]$ can be transferred into $M_0^T(p)=[1\ (0\ 0\ 0)\ (0\ 0)\ (0\ 0)]$. Similarly, $M_1^T(P)$, $M_2^T(P)$, $M_{31}^T(P)$ and $M_{32}^T(P)$ can be decomposed into places [0 (1 1 1) (1 1) (0 0)], [0 (1 1 1) (0 0) (1 1)], [1 (0 0 0) (0 0) (0 0)] and [1 (0 0 0) (0 0) (0 0)], respectively.

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

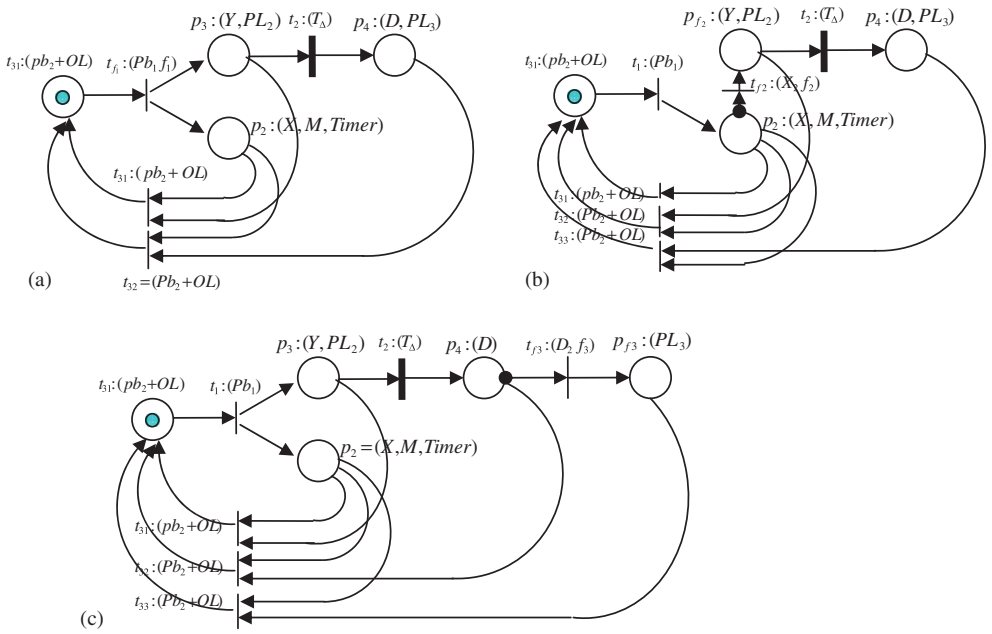


Fig. 12. Petri nets model: (a) with fault f_1; (b) with fault f_2; and (c) with fault f_3 in Fig. 7.

4.4 Analysis and diagnosis of fault modeling

Assume that the faults f_1, f_2 and f_3 in the LD are stuck at 0 (s-a-0), as Fig. 7 illustrates. The fault can then be modeled into Petri nets as shown in Fig. 12(a), (b), and (c), respectively. In case 1, a transition $t_1:(pb_1)$ is fired since pb_1 is active. However, the transition $t_{f_1}:(pb_1 f_1)=0$ cannot be fired since f_1 is stuck at 0. Similarly, in case 2, $t_{f_2}:(X_2 f_2)=0$, $t_2:(X_2)=1$. In case 3, $t_{f_3}:(D_2 f_3)=0$, $t_3:(D_2)=1$. For simple calculation of state equation, a control vector U_k contains the fault free Boolean equation and faulty Boolean equation of transitions, as denoted by $t_{f_i}:(fault\ free/fault)=1/0$. Fig. 13 shows fault free model and fault model simulated structure.

A difference output vector (DOV) = fault free output vector-fault output vector. If has fault occur then difference output vector $\neq 0$. The fault is covered area from place of negative value to place of positive value, and the faulty path flows through transition t_i in DOV.

Case 1. Assume f_1 is s-a-0.

$$A_{f_1}^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_{31} & t_{32} \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{array},$$

$$M_0 = \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, U_1 = \begin{array}{c} t_{f_1}:(pb_1/pb_1 f_1) \\ t_2 \\ t_{31} \\ t_{32} \end{array} \begin{bmatrix} 1/0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$M_{f_1} = M_0 + A_{f1}^T U_1 = \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{array}{c} \\ -1 \\ 1 \\ 1 \\ 0 \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_{31} & t_{32} \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{array} \begin{bmatrix} 1/0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1/0 \\ 1/0 \\ 1/0 \\ 0 \end{bmatrix} = \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \begin{bmatrix} 0/1 \\ 1/0 \\ 1/0 \\ 0 \end{bmatrix}$$

DOV $=[-1\ \ 1\ \ 1\ \ 0]^T$. The faulty area is covered from p_1 to p_2 and p_3 as Fig. 14 indicates, and the fault path flows through $t_1:(pb_1)$. Thus, the fault is located

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

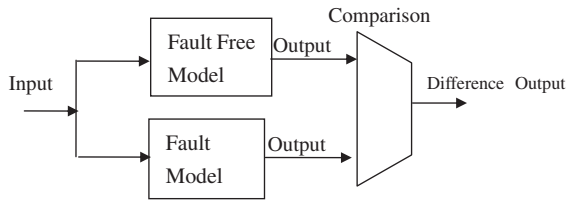


Fig. 13. Simulated fault free model and fault model.

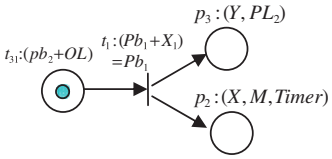


Fig. 14. The faulty area in case 1.

between rung 1 and rung 3 in Fig. 7. In physical terms, this means the motor cannot start rotation.

Case 2. Assume f_2 is s-a-0.

$$A_{f2}^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_{f_2} \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_{f_2} \end{array} \\ \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}; \quad M_0 = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};$$

$$U_1 = \begin{array}{c} t_1:(Pb_1) \\ t_{f_2} \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix};$$

$$U_1 = \begin{array}{c} t_1 \\ t_{f_2}:(X_2/X_2 f_2) \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$M_1(f_2) = M_0 + A_{f2}^T U_1 = \begin{array}{c} p_1 \\ p_2 \\ p_{f_2} \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{array}{c} p_1 \\ +\ p_2 \\ p_{f_2} \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_{f_2} \end{array} \\ \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$M_2(f_2) = M_1 + C_{f2}^T U_2 = \begin{array}{c} p_1 \\ p_2 \\ p_{f_2} \end{array} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{array}{c} p_1 \\ +\ p_2 \\ p_{f_2} \end{array} \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1/0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/0 \end{bmatrix} = \begin{array}{c} p_1 \\ p_2 \\ p_{f_2}:(Y, PL_2) \end{array} \begin{bmatrix} 0 \\ 1 \\ 1/0 \end{bmatrix}$$

DOV $= [0\ \ 0\ \ 1]^T$. The faulty area is covered only in p_{f_2} as Fig. 12b indicates, and the fault path flows through $t_{f_2}:(X_2)$. Thus, the fault is located between rung 2 and rung 5 in Fig. 7. In physical terms, this means the motor cannot run.

Case 3. Assume f_3 is s-a-0.

$$A_{f_3}^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_{f4} \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_{31} & t_{32} & t_{33} & t_{f_3} \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & -1 & -1 & -1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array};$$

$$U_1 = \begin{array}{c} t_1(Pb_1) \\ t_2 \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{f_3} \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad U_2 = \begin{array}{c} t_1 \\ t_2(T_\Delta) \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{f_3} \end{array} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};$$

$$U_{f_3} = \begin{array}{c} t_1 \\ t_2 \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{f_3}:(D_2/D_2 f_3) \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/0 \end{bmatrix}$$

$$M_0 = \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_{f_3} \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad M_1 = M_0 + A_{f_3}^T U_1$$

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

$$\begin{matrix} p_1 \\ p_2 \\ = \ p_3 \\ p_4 \\ p_{f_3} \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \quad M_2 = M_1 + A_{f_3}^T U_2$$

$$\begin{matrix} p_1 \\ p_2 \\ = \ p_3 \\ p_4 \\ p_{f2} \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}; \quad M_{f_3} = M_2 + A_{f_3}^T U_{f_3}$$

$$\begin{matrix} p_1 \\ p_2 \\ = \ p_3 \\ p_4 \\ p_{f_3}:(PL_3) \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1/0 \end{bmatrix}$$

DOV $= [0\ 0\ 0\ 0\ 1]^T$. The faulty area is covered only in p_{f_3} as Fig. 12c indicates, and the fault path flows through $t_{f_3}:(D_2)$. Thus, the fault is located between rung 7 and rung 9 in Fig. 7. In physical terms, this means the indicator light PL_3 cannot light.

V. CONCLUSIONS

This paper proposes the Boolean Petri net using the Boolean equation, constructs a ladder diagram module using the Petri net and develops an abstract model using Petri nets to diagnose local faults in the LD. The diagnostic process employs simple matrix manipulation and DOV to determine the faulty area for diagnosing the ladder diagram. This study also provides an example using composite transition, composite place, and relevant state to reduce complexity and increase readability of the Petri nets. The proposed methodology is useful and clear.

REFERENCES

1. Dotoli, M. G., M. P. Fanti, G. G. Iacobellis, and A. M. Mangini, "A first-order hybrid Petri net model for supply chain management," *IEEE Trans. Autom. Sci. Eng.*, Vol. 6, No. 4, pp. 744–758 (2009).
2. Hu, H. S., M. C. Zhou, and Z. W. Li, "Liveness enforcing supervision of video streaming systems using nonsequential Petri nets," *IEEE Trans. Multimedia*, Vol. 11, No. 8, pp. 1457–1465 (2009).
3. Lan, J. C. and M. Ma, "Fault diagnosis method of power system based on the adaptive fuzzy Petri net," *2009 IEEE Circuits Syst. Int. Conf. Test. Diagn.*, Chengdu, China, pp. 1–4 (2009).
4. Lo, K. L., H. S. Ng, and J. Trecat, "Power systems fault diagnosis using Petri nets," *IEE Proc. C. Gener., Transm. Distrib.*, Vol. 144, pp. 231–236 (1997).
5. Jackman, J., R. Linn, and D. Hyde, "Petri net modeling of relay ladder logic," *J. Des. Manuf.*, Vol. 5, pp. 143–151 (1995).
6. Lee, G. S. and J. S. Lee, "The state equation of Petri net for the LD program," *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Nashville, Tennessee, pp. 3051–3056 (2000).
7. Peng, S. S. and M. C. Zhou, "Ladder diagram sand Petri-net-based discrete-event control design methods," *IEEE Trans. Syst. Man Cybern. Part C—Appl. Rev.*, Vol. 34, No. 4, pp. 523–531 (2004).
8. Venkatesh, K., M. C. Zhou, and R. J. Caudill, "Comparing ladder logic diagrams sand Petri nets for sequence controller design through a discrete manufacturing system," *IEEE Trans. Ind. Electron.*, Vol. 41, No. 6, pp. 611–619 (1994).
9. Venkatesh, K., M. C. Zhou, and R. J. Caudill, "Evaluating the complexity of Petri nets sand ladder logic diagrams for sequence controllers design in flexible automation," *Proc. IEEE Workshop Emerg. Technol. Factory Autom.*, Tokyo, Japan, pp. 428–435 (1994).
10. Lee, J. S., "Design of the remote supervision system for automated processes via the Petri nets approach," Ph.D. Dissertation, National Chiao-Tung University, Taiwan (2004).
11. Murata, T., "Petri nets: properties, analysis, and application," *Proc. IEEE*, Vol. 77, No. 4, pp. 541–580 (1989).
12. Murata, T., "State equation, controllability, and maximal matchings of Petri nets," *IEEE Trans. Autom. Control*, Vol. 22, pp. 412–416 (2007).
13. Zhou, M. C. and E. Twiss, "Design of industrial automated systems via relay ladder logic programming and Petri nets," *IEEE Trans. Syst., Man Cybern. Part C—Appl. Rev.*, Vol. 28, pp. 137–150 (1998).
14. Bender, D. F., B. Combemale, X. Crégut, J. Farines, M. B. Berthomieu, and F. Vernadat, "Ladder metamodeling & PLC program validation through

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

time Petri nets," *ECMDA Lecture Notes Computer Science*, Springer, Berlin/Heidelberg, pp. 121–136 (2008).
15. Abramovici, M., M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, New York (1990).
16. David, R. and H. Alla, *Petri Nets and Grafcet*: *Tools for Modelling Discrete-Event Systems*, Pretrice-Hall, London (1992).

Jui-I Tsai was born in Yunlin, Taiwan, in 1958. He received his B.S. degree in 1985 from the Department of Electronic Engineering, Feng Chia University, Taichung, and M.S. degree in 1994 from the Institute of Medical Engineering, National Cheng Kung University, Taiwan. He is a Ph.D. candidate at Department of Electric Engineering, National Chiao Tung University, Hsinchu. His research interests include Petri nets and IC testing.

Ching-Cheng Teng was born in Taiwan, in 1938. He received the B.S. degree in Electric Engineering from the Nation Cheng Kung University, Taiwan, in 1961.

From 1991 to 1998, he was the chairman of the Department of Electrical and Control Engineering, Nation Chiao Tung University, Hsinchu, Taiwan. He is currently a professor in the Department of Electric Engineering at the same university. His research interests include H^{∞} optimal control, signal processing, and fuzzy neural systems.

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society