Generalized State Equation of Petri Nets with Priority

Gi Bum Lee,[1,*] Han Zandong,[2,†] Jin S. Lee[3,‡]
[1]*Research Institute of Industrial Science and Technology, Pohang, 790-600, Korea*
[2]*Tsinghua University, Beijing 100084, P.R. China*
[3]*Pohang University of Science and Technology, Pohang, 790-784, Korea*

This article presents a new way of generating a generalized state equation that is useful for analyzing the token flow of the Petri Net (PN) with priority. The transition values in the firing vector as used in the conventional state equation are replaced with transition variables, which are generated by multiplying a series of firing condition functions taking the weighted inhibitor arc into account. The actual value of a transition variable is determined by taking priority and the present marking into account. The proposed state equation generalizes the conventional one by using the transition variable form and by containing the formulation of priority. Given the initial marking, the subsequent marking evolution can be determined successively from the generalized state equation as the simultaneous firings occur. A PN with deadlock is analyzed as an example to establish the validity of the generalized state equation. © 2003 Wiley Periodicals, Inc.

1. INTRODUCTION

Petri Net[1,2] (PN) is a general-purpose graphical and mathematical tool for discrete event systems and provides a uniform environment for modeling, formal analysis, and design. As a mathematical tool, the PN describes the model either by using a set of linear algebraic equations or by using other mathematical approaches. This makes the formal analysis of the model possible, thereby verifying the behavior of the underlying system, e.g., precedence relations among events, concurrent operations, appropriate synchronization, freedom from deadlock, repetitive activities, and mutual exclusion of shared resources.[3–5] The ability of PNs to verify formally the model is especially important for the real-time safety-critical automated systems.

Recently, a number of results on the state equation formulation of PNs have been reported in the literature. Iko Miyazawa et al.[6] used the state equation to solve

*Author to whom all correspondence should be addressed: e-mail: gblee@rist.re.kr.
†e-mail: hanzd@mail.tsinghua.edu.cn.
‡e-mail: jsoo@postech.ac.kr.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 18, 1145–1153 (2003)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.10135

the reachability problem of PNs with known firing sequence. Burns and Bidanda used the concept of transition variables to translate safe PNs into sequential Boolean equations but not to formulate the state equation.[7] Chamas and Singh extended Murata's state equations for standard PNs to other continuous time systems; however, the involved equations are not linear equations but differential equations.[8] Matsumoto and Miyano have presented a formal necessary and sufficient condition on reachability of general PNs under the condition that a nonnegative integer solution of the state equation has been given with the known firing count vectors.[9] Murata presented PNs as discrete time systems and analyzed the controllability and reachability in terms of the matrix representation of PNs.[10]

Most of these studies used the incidence matrix and the firing vector to formulate a state equation. On the contrary, this study introduces a way of generating a generalized state equation of PNs with transition variables formed with firing condition functions. The transition values in the firing vector as used in the conventional state equation are replaced with the transition variables, which are generated by multiplying a series of firing condition functions. If a conflict occurred by means of simultaneous firing, transitions associated with the conflict are assigned manually to priority in order to solve the conflict problem.[11,12] However, the formulation to calculate the firing state of the conflict transitions with respect to priority is created and contained in the generalized state equation proposed in this article. Given the initial marking, the subsequent marking evolution then is determined easily by analyzing the proposed state equation containing priority. A token flow of PN with deadlock is analyzed according to priority as an example to establish the validity of the proposed state equation.

2. GENERALIZED STATE EQUATION WITH FIRING CONDITION FUNCTIONS

A PN is defined as a five-tuple PN $= (P, T, I, O, M_0)$, where $P = \{P_1, P_2, \cdots P_n\}$ is a finite set of places; $T = \{T_1, T_2, \cdots T_m\}$ is a finite set of transitions $P \cup T \neq \phi$, and $P \cap T = \phi$; $I \subseteq (P \times T) \rightarrow N$ is the input function that defines the set of directed arcs from places to transitions, where N is a set of nonnegative integers; $O \subseteq (T \times P) \rightarrow N$ is the output function that defines the set of directed arcs from transitions to places; $M_0 = \{m_1, m_2, \cdots, m_n\}$ is the initial marking.

The state equation that describes the correlations between places, transitions, and arcs is instrumental in analyzing the properties of the PN graph. In formulating the state equation, $\mathbf{P} \in \mathbf{R}^{n\times 1}$ is defined as the vector representing marking and $\mathbf{T} \in \mathbf{R}^{m\times 1}$ is defined as the firing vector representing transition values, which is one when an associated transition is enabled and zero otherwise. An ith element of $\mathbf{P}$ is denoted by P_i $(i = 1, 2, \ldots, n)$ and a jth element of $\mathbf{T}$ is denoted by T_j $(j = 1, 2, \ldots, m)$. A place P_i is an input place of a transition T_j if $P_i \in I(T_j)$; P_i is an output place of T_j if $P_i \in O(T_j)$. Consequently, the relation of the input and output function in a PN graph is shown in Figure 1.

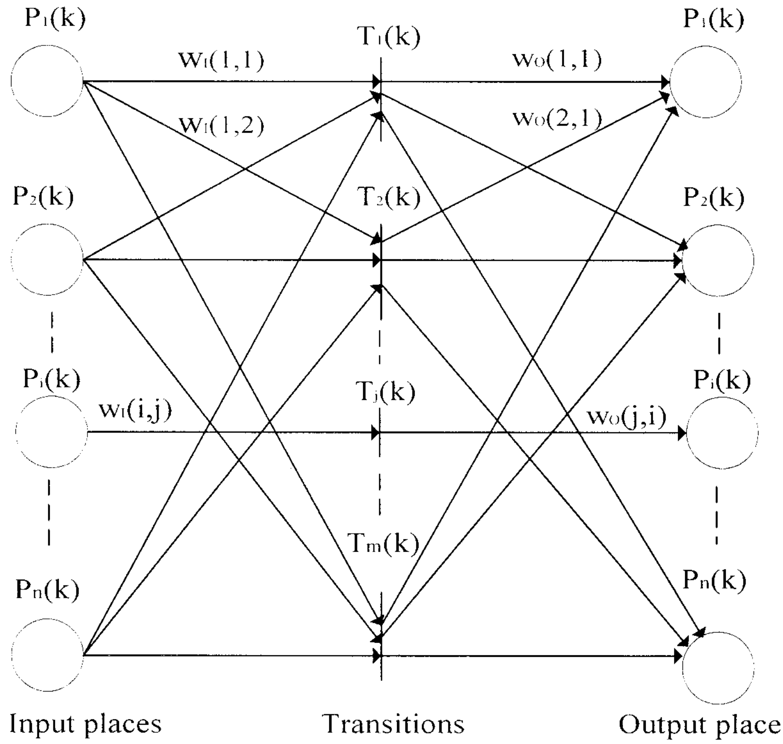

Figure 1. Relation of input and output function in a PN graph.

Now, we define, respectively, the firing vector **T** and the marking vector **P** by

$$\mathbf{T} = [T_1 \quad T_2 \quad \cdots \quad T_m]^T \tag{1}$$

$$\mathbf{P} = [P_1 \quad P_2 \quad \cdots \quad P_n]^T \tag{2}$$

and we define the incidence matrix $\mathbf{J} \in \mathbf{R}^{n\times m}$ by

$$\mathbf{J} = \mathbf{O} - \mathbf{I} = \begin{bmatrix} O_{11} & \cdots & O_{1m} \\ \vdots & & \vdots \\ O_{n1} & \cdots & O_{nm} \end{bmatrix} - \begin{bmatrix} I_{11} & \cdots & I_{1m} \\ \vdots & & \vdots \\ I_{n1} & \cdots & I_{nm} \end{bmatrix} = \begin{bmatrix} J_{11} & \cdots & J_{1m} \\ \vdots & & \vdots \\ J_{n1} & \cdots & J_{nm} \end{bmatrix} \tag{3}$$

where

$$O_{ij} = \begin{cases} w_O(j, i), & \text{if } \{(T_j, P_i) \in R\} \\ 0, & \text{otherwise} \end{cases}$$

$$I_{ij} = \begin{cases} w_I(i, j), & \text{if } \{(P_i, T_j) \in R\} \\ P_i(k), & \text{if } \{(P_i, T_j) \in V\} \\ 0, & \text{otherwise} \end{cases}$$

The $(x, y) \in R$ implies that there is an ordinary arc from x to y; $(x, y) \in V$ implies that there is a *reset arc* from x to y. If there is a reset arc between P_i and

T_j, all tokens in P_i are removed when T_j fires. The term $P_i(k)$ is the variable that is associated with the number of tokens in place P_i at sampling step k; $w_I(i, j)$ and $w_O(j, i)$ are the weights of the input and output arcs, respectively. Note here that a reset arc is taken into account in generating the incidence matrix.

A transition T_j is enabled if the following three conditions are satisfied:

(A) For each ordinary input place P_i, $M(P_i) \geqslant w_I(i, j)$
(B) For each inhibitor input place P_i, $M(P_i) < w_I(i, j)$
(C) For each reset input place P_i, $M(P_i) \geqslant 1$

Here, $M(P_i)$ represents the number of tokens in P_i. The firing of T_j does not change the token number of its inhibitor input place.

At each sampling step k, the state equation in conventional form then is written as

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \mathbf{J} \cdot \mathbf{T}(k) \tag{4}$$

where $\mathbf{J} \cdot \mathbf{T}(k)$ shows the change of the tokens under simultaneous firing; $\mathbf{P}(k+1)$ is a column vector of the next state; $\mathbf{P}(k)$ is a column vector of the present state.

To enable transition, all of the input places must satisfy the enable conditions. The state equation in the generalized form is created taking the firing condition into account. First, the transition variable T_j for $j = 1, 2, \cdots, m$ is generated as

$$T_j(k) = d_j^1(k) \cdot d_j^2(k) \cdot d_j^3(k) \cdots d_j^n(k) = \prod_{i=1}^{n} d_j^i(k) \tag{5}$$

where $d_j^i(k)$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$ is a *firing condition function* that is associated with P_i and T_j. It is defined as

$$d_j^i(k) = \begin{cases} \mathcal{T}\left(\dfrac{P_i(k)}{I_{ij}}\right), & \text{if } (P_i, T_j) \in R \\ \mathcal{N}\left(\dfrac{P_i(k)}{w_I(i,j)}\right), & \text{if } (P_i, T_j) \in Q \\ \mathcal{T}(P_i(k)), & \text{if } (P_i, T_j) \in V \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where

$$\mathcal{T}(x) = \begin{cases} 1, & \text{if } x \geqslant 1 \\ 0, & \text{if } x < 1 \end{cases} \quad \text{and} \quad \mathcal{N}(x) = \begin{cases} 1, & \text{if } x < 1 \\ 0, & \text{if } x \geqslant 1 \end{cases}$$

and Q is a weighted inhibitor arc[13] from x to y. The firing condition function is used to determine under which condition the transition is enabled on the arc that connects P_i to T_j. Consequently, substituting Equation 5 into Equation 4 gives the *generalized state equation* containing firing condition functions:

$$\begin{bmatrix} P_1(k+1) \\ P_2(k+1) \\ \vdots \\ P_i(k+1) \\ \vdots \\ P_n(k+1) \end{bmatrix} = \mathbf{P}(k) + \mathbf{J} \cdot \begin{bmatrix} T_1(k) \\ T_2(k) \\ \vdots \\ T_j(k) \\ \vdots \\ T_m(k) \end{bmatrix} \tag{7}$$

Using Equation 7, we calculate $\mathbf{P}(k+1)$ and check if any of its component becomes negative. If $P_i(k+1)$, the ith component of $\mathbf{P}(k+1)$, is negative, then there is a conflict associated with place P_i. The output transitions of P_i that correspond to negative values in the ith row of the incidence matrix $\mathbf{J}$ are the conflict transitions, which are collected and marked as $(T_{c1}, \cdots, T_{cg}) \in P_i^*$, where P_i^* is the set of output transitions of P_i. In the presented algorithm, we assign the priority sequence as $T_{c1} > \cdots > T_{cg}$ and set

$$T_j = T_{cr}, \quad \text{where} \quad r = 1, 2, \ldots, g, T_j \in P_i^* \tag{8}$$

The subscript of $T_j = T_{cr}$ then is copied into v_r as

$$v_r = j, \quad \text{where} \quad r = 1, 2, \ldots, g \tag{9}$$

Each transition value for conflict transitions with priority can be calculated as follows:

$$T_{c1}(k) = \mathcal{T}\left(\frac{P_i(k)}{w_I(i, v_1)}\right) \tag{10}$$

$$T_{cr}(k) = \mathcal{T}\left(\frac{P_i(k) - \sum_{q=1}^{r-1} w_I(i, v_q) T_{cq}(k)}{w_I(i, v_r)}\right), \quad \text{for} \quad r = 2, 3, \ldots, g \tag{11}$$

The resulting values of Equations 8, 10, and 11 are substituted into Equation 7 and the state equation is solved once again at the same sampling step k.

3. MARKING ANALYSIS BASED ON THE PROPOSED STATE EQUATION

As an example of using the proposed state equation, we introduce a PN graph as shown in Figure 2 and analyze the marking evolution taking the priority into account. Once the initial marking is assigned, the successive markings are computed sequentially from Equation 7. We give the initial marking $M_0 = (3, 0, 0, 0)$. The incidence matrix of the PN graph is computed as

$$\mathbf{J} = \mathbf{O} - \mathbf{I} = \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{12}$$

and the state equation with transition variables is expressed as

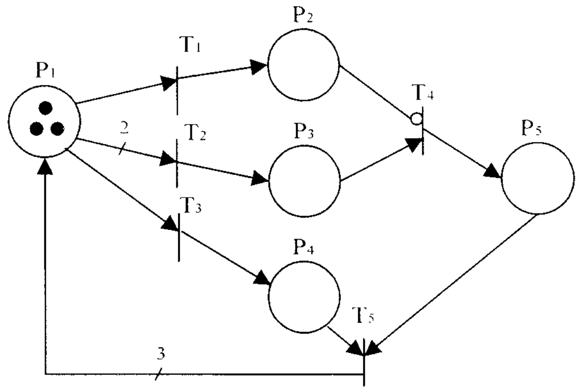

Figure 2. Weighted inhibitor arc PN with its initial marking.

$$\begin{bmatrix} P_1(k+1) \\ P_2(k+1) \\ P_3(k+1) \\ P_4(k+1) \\ P_5(k+1) \end{bmatrix} = \begin{bmatrix} P_1(k) \\ P_2(k) \\ P_3(k) \\ P_4(k) \\ P_5(k) \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$
$$\times \begin{bmatrix} d_1^1(k) \cdot d_1^2(k) \cdot d_1^3(k) \cdot d_1^4(k) \cdot d_1^5(k) \\ d_2^1(k) \cdot d_2^2(k) \cdot d_2^3(k) \cdot d_2^4(k) \cdot d_2^5(k) \\ d_3^1(k) \cdot d_3^2(k) \cdot d_3^3(k) \cdot d_3^4(k) \cdot d_3^5(k) \\ d_4^1(k) \cdot d_4^2(k) \cdot d_4^3(k) \cdot d_4^4(k) \cdot d_4^5(k) \\ d_5^1(k) \cdot d_5^2(k) \cdot d_5^3(k) \cdot d_5^4(k) \cdot d_5^5(k) \end{bmatrix} \tag{13}$$

where

$$d_1^1(k) = \mathcal{T}\left(\frac{P_1(k)}{1}\right), \quad d_1^2(k) = d_1^3(k) = d_1^4(k) = d_1^5(k) = 1$$

$$d_2^1(k) = \mathcal{T}\left(\frac{P_1(k)}{2}\right), \quad d_2^2(k) = d_2^3(k) = d_2^4(k) = d_2^5(k) = 1$$

$$d_3^1(k) = \mathcal{T}\left(\frac{P_1(k)}{1}\right), \quad d_3^2(k) = d_3^3(k) = d_3^4(k) = d_3^5(k) = 1$$

$$d_4^1(k) = 1, \quad d_4^2(k) = \mathcal{N}\left(\frac{P_2(k)}{1}\right), \quad d_4^3(k) = \mathcal{T}\left(\frac{P_3(k)}{1}\right), \quad d_4^4(k) = d_4^5(k) = 1$$

$$d_5^1(k) = d_5^2(k) = d_5^3(k) = 1, \quad d_5^4(k) = \mathcal{T}\left(\frac{P_4(k)}{1}\right), \quad d_5^5(k) = \mathcal{T}\left(\frac{P_5(k)}{1}\right)$$

3.1. Marking Analysis Without Priority

When the first simultaneous firing occurs without priority, the following marking becomes

$$\begin{bmatrix} P_1(1) \\ P_2(1) \\ P_3(1) \\ P_4(1) \\ P_5(1) \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{14}$$

which shows that $P_1(1) < 0$ and the place P_1 generates conflict and the marking evolution can no longer continue. Then, the conflict transitions of P_1 can be found from the first row of the incidence matrix with negative values, which is T_1, T_2, and T_3.

3.2. Marking Analysis with Priority (I)

We assign priority to the conflict transitions in such a way that $T_1 = T_{c1}$, $T_2 = T_{c2}$, and $T_3 = T_{c3}$, and then $v_1 = 1$, $v_2 = 2$, and $v_3 = 3$. According to the assigned priority, each transition value can be calculated from Equations 8, 10, and 11, which becomes

$$\begin{aligned} T_1 &= T_{c1} = 1 \\ T_2 &= T_{c2} = 1 \\ T_3 &= T_{c3} = 0 \end{aligned} \tag{15}$$

The values of T_1, T_2, and T_3 then are substituted into Equation 13 and the remaining transition values (T_4, T_5) can be computed by using the firing condition function. Then, we apply the simultaneous firing again and we have from Equation 13 that

$$\begin{bmatrix} P_1(1) \\ P_2(1) \\ P_3(1) \\ P_4(1) \\ P_5(1) \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

When the subsequent simultaneous firing is applied, the consequent marking at every place becomes

$$\begin{bmatrix} P_1(2) \\ P_2(2) \\ P_3(2) \\ P_4(2) \\ P_5(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

Notice here that the resulting marking will never be changed even if the computation is repeated. Any transition can not be fired at this moment because the transition vector turned out to be $\mathbf{T} = [0 \quad 0 \quad 0 \quad 0 \quad 0]$ in Equation 17. This is the deadlock situation.

3.3. Marking Analysis with Priority (II)

Now, we assign another priority to transitions in such a way that $T_3 = T_{c1}$, $T_2 = T_{c2}$, and $T_1 = T_{c3}$, and then $v_1 = 3$, $v_2 = 2$, and $v_3 = 1$. According to the assigned priority, each transition is calculated from Equations 8, 10, and 11, which becomes

$$\begin{aligned} T_3 &= T_{c1} = 1 \\ T_2 &= T_{c2} = 1 \\ T_1 &= T_{c3} = 0 \end{aligned} \tag{18}$$

The resulting values of T_1, T_2, and T_3 are substituted into Equation 13 and the remaining transition values (T_4, T_5) are calculated by using the firing condition function. Then, applying the simultaneous firing again, it follows from Equation 13 that

$$\begin{bmatrix} P_1(1) \\ P_2(1) \\ P_3(1) \\ P_4(1) \\ P_5(1) \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{19}$$

When the subsequent simultaneous firing is applied, the marking at every place becomes

$$\begin{bmatrix} P_1(2) \\ P_2(2) \\ P_3(2) \\ P_4(2) \\ P_5(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{20}$$

Once again, when the next simultaneous firing is applied again, then the consequent marking at every place becomes

$$\begin{bmatrix} P_1(3) \\ P_2(3) \\ P_3(3) \\ P_4(3) \\ P_5(3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

Notice here that the consequent marking returns to the initial marking. Therefore, the PN is free from deadlock and three-bounded under the given priority.

When the priority is assigned as $T_1 = T_{c1}$, $T_2 = T_{c2}$, and $T_3 = T_{c3}$, the PN is stuck in the deadlock state. However, when the priority is assigned as $T_3 = T_{c1}$, $T_2 = T_{c2}$, and $T_1 = T_{c3}$, PN is free from deadlock. Consequently, the deadlock problem can be solved by assigning reasonable priority to the conflict transition.

4. CONCLUSIONS

This study proposes a way of generating the generalized state equation, which is useful for analyzing PNs with priority. The generalized state equation containing the formulation of priority is represented with the transition variables, which are generated by multiplying a series of firing condition functions. Given the initial marking of the PN graph, the subsequent marking evolution with the simultaneous firing can be deduced easily by using the proposed state equation. An example has shown that deadlock with respect to the simultaneous firing can be avoided by assigning reasonable priority to the conflict transitions.

References

1. Murata T. Petri Nets: Properties, analysis, and applications. Proc IEEE 1989;77:541–580.
2. David R, Alla H. Petri Nets and grafcet. Prentice Hall International (UK) Ltd., 1992.
3. Lewis FL, Gurel A, Bogdan S, Doganalp A, Pastravanu OC. Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems. Automatica 1998;34:1083–1100.
4. Lee GB, Lee JS. The state equation of Petri Net for the LD program. In: IEEE Int Conf on Systems, Man, and Cybernetics, 2000. pp 3051–3056.
5. Zhou MC. Petri Nets in flexible and agile automation. Boston/Dordrecht/London: Kluwer Academic Publishers; 1995.
6. Miyazawa I, Tanaka H, Sekiguchi T. Classification of solutions of matrix equation related to parallel structure of a Petri Net. In: IEEE Conf on Emerging Technologies and Factory Automation. 1996. pp 446–452.
7. Burns GL, Bidanda B. The use of hierarchical Petri Nets for the automatic generation of ladder logic programs. In: Proc Of ESD IPC-94 Conf and Exposition, Detroit, MI, 1994. pp 169–179.
8. Chamas N, Singh H. A generalized Petri Net state equation. In: Midwest Symposium on Circuits and System. 1990. pp 161–163.
9. Matsumoto T, Miyano Y. Reachability criterion for Petri Nets with known firing count vectors. IEICE Trans Fund 1998;E81-A:628–634.
10. Murata T. State equation, controllability, and maximal matchings of Petri Nets. IEEE Trans Automatic Control 1977;22:412–416.
11. Hack M. Decidability questions for Petri Nets. Ph.D. dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1975.
12. Peterson, JL. Petri Net theory and the modeling of systems. Englewood Cliffs, NJ: Prentice-Hall Inc., 1981.
13. Uzam M, Jones AH. Discrete event control system design using automation Petri Nets and their ladder diagram implementation. Int J Adv Manufact Technol 1998;14:716–728.