Fuzzy Optim Decis Making (2008) 7:399–408
DOI 10.1007/s10700-008-9047-9

The design of a controller in Fuzzy Petri net

Yubin Zhong

Published online: 30 September 2008

Abstract This paper designs a class of Fuzzy Petri net controller based on the character of place invariant in Petri net. This controller is simple in operation and is easy to put into practice. It has satisfying effects and can provide methods to do research in designing and optimizing the performance of system in solving the dead lock phenomenon in concurrent system.

Keywords Place invariant · Controller · Fuzzy Petri net · Dead lock phenomenon · Optimize

1 Introduction

Having strict mathematical formulation, intuitive graphical expression, rich description way of the system and behavior analysis technology, providing a solid conceptual basis for computer science, Petri net, which is proposed and invented in the 1960s, is the discrete mathematics representation of concurrent system. It is suitable for the description of asynchronous and concurrent computer system model. Since Petri net can be expressed as the concurrent incident, it is also regarded as a theory of automation. Research areas tend to think that Petri net is the original of all process definition language.

Petri net is an important implement in building models and analyzing discrete events dynamic systems. It was widely used in controller design and in building models for many systems. Further more, it does not only help to research in the performance of systems, but also controls and optimizes it based on models. However, the models built

Y. Zhong (✉)
School of Mathematics and Information Sciences, Guangzhou University, Guangzhou, Guangdong 510006, People's Republic of China
e-mail: Zhong_yb@163.com

were normally huge because of the complexity of the real system. So if general Petri net is applied in analyzing the performance of system directly, it can lead to dimension disaster on computer. In order to solve this problem, this article proposes a new controller design based on Petri net (Looney 1988), which generalizes control designing (Holloway and Krogh 1990; Wang and Zhu 1998; Zhou et al. 1992) to Fuzzy Petri net. A new design method for maximum feedback allowed controller (Chen and Zhong 2003) is given. This controller is composed of variance place which is connected to Fuzzy Petri net. The dead lock phenomenon can be solved and can ensure system to avoid access in some forbidden state according to given constraints if this method is put to use.

2 The definition of classical Petri net

Composed of place, transition, connection and token, classical Petri net is a simply process model.

2.1 Structure of Petri net

(1) Element of Petri Net
Place is represented as round node
Transition is represented as quadrate node
Connection is a directed arc between place and transition
Token is the dynamic object of place, which can move from one place to another
(2) Rules of Petri Net
Connection is directed
There are no arcs between two places or transitions
One Place may have numbers of tokens

2.2 Behavior of Petri net

If every input place of a transition has a token, then it is set to be enabled and fired, the token of input place is consumed while a new token is produced for input palace.

Every transition is atomic.

Two transitions are possibly to be accepted, however, only one ofthem fires at a time.

Petri net is static.

State of Petri net depends on the distribution of place's token.

2.3 Formatted specification of Petri net

A classical Petri net is composed of an array which has four element, they are place, transition, input function and output function. Any diagram can be mapped to the kind of array and it is the same conversely.

2.4 Petri net process modeling

The state of a flow is modeling by token of place, transition of state is modeling by transition. Token represents something like person, goods, machine, information, conditions or state of object. Place represents aisle or geography location. Transition represents incident, changing or transportation. A flow have three states, they are current state, reachable state and unreachable state.

2.5 Limitation of classical Petri net

Not be able to test zero token in place
Tend to become very large model
Does not reflect the contents of the time
Not supported to construct large scale of modeling, such as from top to bottom or from bottom to top.

3 The definition of Fuzzy Petri net

3.1 The definition of Fuzzy Petri net

Fuzzy Petri net (FPN for short) is a knowledge model, which is based on traditional Fuzzy generating regulation and Petri net and is composed of place, variance, reliability, and valve value.

FPN is defined as an eight-tuple, FPN $=$ (P, T, I, O, C_f, a, b, M_0), in this eight-tuple p $= (p_1, p_2, \ldots, p_3)$ is limited place set; T $= (t_1, t_2, \ldots, t_n)$ is limited variance set, P $\cap$ T $= \Phi$; I is input function, place to directed arc set of variance $I(t_j) = p_i$; O is output function, the varied place directed arcs set O $(p_i) = t_j$; C_f:T$\rightarrow$[0,1] is the reliability of variance; a: P $\rightarrow$[0,1] is reliability of place; b: T $\rightarrow$[0,1] is valve value which is activated in varying. About $p_i \in I(t_j)$, if $a(p_i) > b(t_j)$, the variance t_j can start up, M_0 is the initial Fuzzy token.

3.2 The rules of activation

In Fuzzy Petri net, in order to activate the variant, the following two conditions must be satisfied:

a. There must exist a token, which represents a Fuzzy inputting variable in the place where you input;
b. It must satisfy Fuzzy regulation condition that associates with variant, that reliability in inputting place must be greater than its valve value.

3.3 Place invariant

Based on one of the structure characteristics of Petri net, which depends on its topological structure but not its initial label, place invariant is the place set that holds a fixed number of tokens and it can be expressed by an n-dimensional column vector X. Its nonzero elements are corresponding to special invariant places, while zero elements are on the contrary. A place invariant is defined as an integer vector X, and satisfies:

$$\mathrm{M}^T\mathrm{X} = \mathrm{M}_0^T\mathrm{X} \tag{1}$$

M_0 is the initial label of the network, M is the next label, the meaning of Eq. 1 is that in invariant place the summation of all weight, which signs its token, is constant and the summation is determined by the initial label of Petri net. Place invariant of network can be deduced by the integral solution of the following equation:

$$\mathrm{M}^T\mathrm{D} = 0 \tag{2}$$

D is connection matrix with $\mathrm{n}\times\mathrm{m}$ dimensions, $\mathrm{D} = \mathrm{I}\ \mathrm{O}$, n and m are the place number and the variant number of network, respectively. We can observe that the arbitrariness linear combination of place invariant is place invariant in network. Place invariant is the key value for analyzing Petri net, because it allows analyzing the structure of network independently and dynamically.

4 Design of controller

Suppose the controlled object system is a Fuzzy Petri net model which is composed of n places and m variants. Here b equal to 0. The aim of controlling is force process to obey the constraint of the following inequality:

$$\mathrm{l}_1\mathrm{M}_i + \mathrm{l}_2\mathrm{M}_j \le \mathrm{b}_j \tag{3}$$

M_i and M_j are the labels of the places p_i and $\mathrm{p}_j$in Fuzzy NET. l_1, l_2 and $\mathrm{p}_j$are integer constants. After laxity variable M_s is brought, the constraint of inequality can be transformed into equation form:

$$\mathrm{l}_1\mathrm{M}_i + \mathrm{l}_2\mathrm{M}_j + \mathrm{M}_s = \mathrm{b}_j \tag{4}$$

M_s represents a new place p_s, it has additional Fuzzy token which is needed to balance two sides of the equation. It ensures that the summation of Fuzzy token weight in place p_i and p_j is less or equal to b_j all the time. The place p_s ought to belong to the controlling network. To calculate its structure we can bring in laxity variable M_s, then bring in a place invariant according to whole controlled system which is defined by Eq. 4. Obviously, the number of constraints in inequality 3 is the same as the number of controller places. So the size of controllers is in proportion to the number of constraints in inequality 3.

Because of bringing in a new place in network, the connection matrix D of controlled system changes into $\mathrm{n} \times \mathrm{m}$ dimensional matrix D_p, Plus one row corresponding to the place brought by the laxity variable M_s. This row, which is called D_s, belongs to the connection matrix of the controller. To calculate the arc which connects controller place to system initial Fuzzy Prtri net, we use Eq. 2 of place invariant. When vector x_j is place invariant defined by Eq. 4, unknown values are the elements of the new row in matrix.

We can describe the above problem as follows: All constraints in inequality (3) can be written into matrix forms:

$$\mathrm{LM}_p \leq \mathbf{B} \tag{5}$$

Here M_p is label vector of Fuzzy Petri net, L is $\mathrm{n}_c \times \mathrm{n}$ dimensional integer matrix, n_c is the number of constraints in inequality (3), B is $\mathrm{n}_c \times 1$ dimensional integer vector.

We can change (5) into equation by bringing in laxity variable:

$$\mathrm{LM}_p + \mathrm{M}_c = \mathbf{B} \tag{6}$$

M_c is $\mathrm{n}_c \times 1$ dimensional integer vector which indicates controlling place label. Because every place invariant defined by Eq.6 must satisfy Eq.2, we can deduce:

$$\begin{aligned} \mathrm{X}^T \mathrm{D} &= 0 \\ [\mathrm{L},\mathrm{I}][\mathrm{D}_p, \mathrm{D}_c]^T &= 0 \end{aligned} \tag{7}$$

Here I is $\mathrm{n}_c \times \mathrm{n}_c$ dimensional identity matrix, because the coefficients of laxity variable are1, it includes arcs which connect controlling place to Fuzzy Petri net in matrix D_c. So according to the given Fuzzy Petri net model D_p and constraints L and B which process must be satisfied, Fuzzy Petri net controller D_c can be defined as follows:

$$\mathrm{D}_c = -\mathrm{LD}_p \tag{8}$$

We must notice that the initial label in controller should be included. The initial label of controller Mc_0 is the label that can satisfy place invariant Eq. 6 and depends on the initial label of Fuzzy Petri net. As to the Given the Eq. 1, initial label vector Eq. 6 can be defined as following form:

$$\mathrm{LM}\ \mathrm{p}_0 + \mathrm{Mc}_0 = \mathbf{B} \tag{9}$$

Which is

$$\mathrm{Mc}_0 = \mathrm{B}\text{-}\mathrm{LMp}_0 \tag{10}$$

Below is an example of controller design, as shown in Fig. 1. The connection matrix of network is:

$$D_p = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

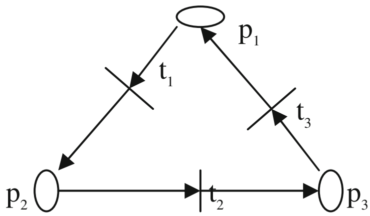

Fig. 1 Primary Petri net

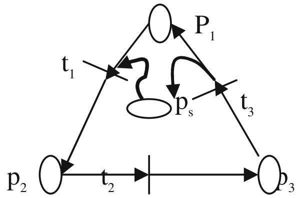

Fig. 2 Network contain controller

Its initial Fuzzy label is:

$$\mathrm{Mp}_0 = [\mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3]^T = [1,0,0]^T$$

Assume the aim of controlling is that place p_2 and p_3 do not contain Fuzzy token at the same time, which is:

$$\begin{aligned} \mathrm{M}_2 + \mathrm{M}_3 &\leq 1 \\ \mathrm{L} = [0,1,1], \mathrm{B} &= l \end{aligned} \tag{11}$$

Because [1,0,0] is not a place invariant of network, the primary network does not satisfy given constraints. we change Eq. 11 into equation constraint by bringing in laxity variable M_s:

$$\mathrm{M}_2 + \mathbf{M}_3 + \mathrm{Ms} = 1 \tag{12}$$

Laxity variable M_s shows Fuzzy label of the place p_s in controller, the label $\mathrm{X}^T = [0,1,1]$ of Eq. 12 is the place invariant of the controlled network. We can calculate the connection matrix of the controlled net by Eq. 8:

$$\mathrm{D}_c = -\mathrm{LD}_p = [-1,0,1]$$

The initial condition of controller place is given by Eq. 10:

$$\mathrm{Mc}_0 = \mathrm{B} - \mathrm{L\ M\ p}_0 = 1$$

Figure 2 shows the place p_s in Petri net and its connection which belongs to feedback controller.

5 A synthetical algorithm aiming at the dead lock phenomenon

Now, we utilize the model of Fuzzy Petri net to provide a synthetical algorithm aiming at the dead lock phenomenon based on Fuzzy Petri net model controller. Detailed steps are:

Step 1: Produce its reachable state fig. RMG(Σ) from Petri net model (Σ) of primary system;
Step 2: Produce reachable state fig. RMG($\Sigma\Theta$C) of object system from reachable state fig. RMG(Σ) and control criterion (Spec)of primary system;
Step 3: Produce reachable state fig. RMG(C) of controller from reachable state fig. RMG($\Sigma\Theta$C) of object system;
Step 4: Produce Petri net model (C)of controller from reachable sate fig. RMG(C)

The detailed processes of every step above are:

Algorithm 1:
Input: RMG($\Sigma\Theta$C) = (V, E, F), output:RMG(C) = (V_c,E_c,F_c)

Step1. Let $V_c = \{\Gamma P \to P_c(M) | (M \in V) \wedge (\Gamma P \to P_c$ (M) is projection sub-vector of M to $P_c\}$;
Step2. Suppose M, $M'' \in$ V, (M,M″) $\in$ E, and f((M, M″)) = t, then we can let $E_c \leftarrow E_c \cup \{(\Gamma P \to P_c$ (M), $\Gamma P \to P_c$ (M″))}, Fc($\Gamma P \to P_c$ (M), $\Gamma P \to P_c$ (M″)) $\leftarrow$t, because $\Sigma\Theta$C is compounded synchronously from Σ and C actually, for t $\in$ T $\Sigma\Theta$C, we have cases as follows:

Case 1 If t $\in$ TΣ-TC, then M[t]>M″, iff $\Gamma P \to P\Sigma$ (M)[t] > $\Gamma P \to P\Sigma$ (M″), and $\Gamma P \to$
P_c (M)= $\Gamma P \to P_c$ (M″);
Case 2 If t $\in$ TΣ-TC, then M [t] > M″, iff $\Gamma P \to P_c$ (M)[t] > $\Gamma P \to$Pc(M″), and $\Gamma P \to P\Sigma$ (M) = $\Gamma P \to P\Sigma$ (M″);
Case 3 If t $\in$ T$\Sigma \cap$TC, then M [t]>M″, iff $\Gamma P \to P\Sigma$ (M)[t] > $\Gamma P \to P\Sigma$ (M″), and $\Gamma P \to P_c$ (M)[t] > $\Gamma P \to P_c$ (M″);

Thus, RMG(C) = (V_c,E_c,F_c) is obtained by RMG($\Sigma\Theta$C) = (V,E,F) projecting in C actually.

Algorithm 2:
Now suppose input: RMG(C), output :C = (P_c,T_c,F_c,Mc_0)

Step 1. Let $F_c \leftarrow$, mark Mc_0, push Mc_0 in stack;
Step 2. If stack is not empty, do Step3. Otherwise finish;
Step 3. If there exists a node M_c, which is adjacent to top but has not been marked, do Step 4. Otherwise go to Step 5;
Step 4. Let ΔM_c = M_c-stack(top), mark f((stack(top), M_c)) = t, do Step 4.1;
Step 4.1 For $p_c \in$ Pc, if $\Delta Mc(p_c) > 0$, then let $F_c \leftarrow F_c \cup \{(t, p_c)\}$; if $\Delta Mc(p_c) <$ 0, then let $F_c \leftarrow F_c \cup \{(p_c,t)\}$,
Step 4.2 Mark M_c and push M_c in stack, go to Step 3;

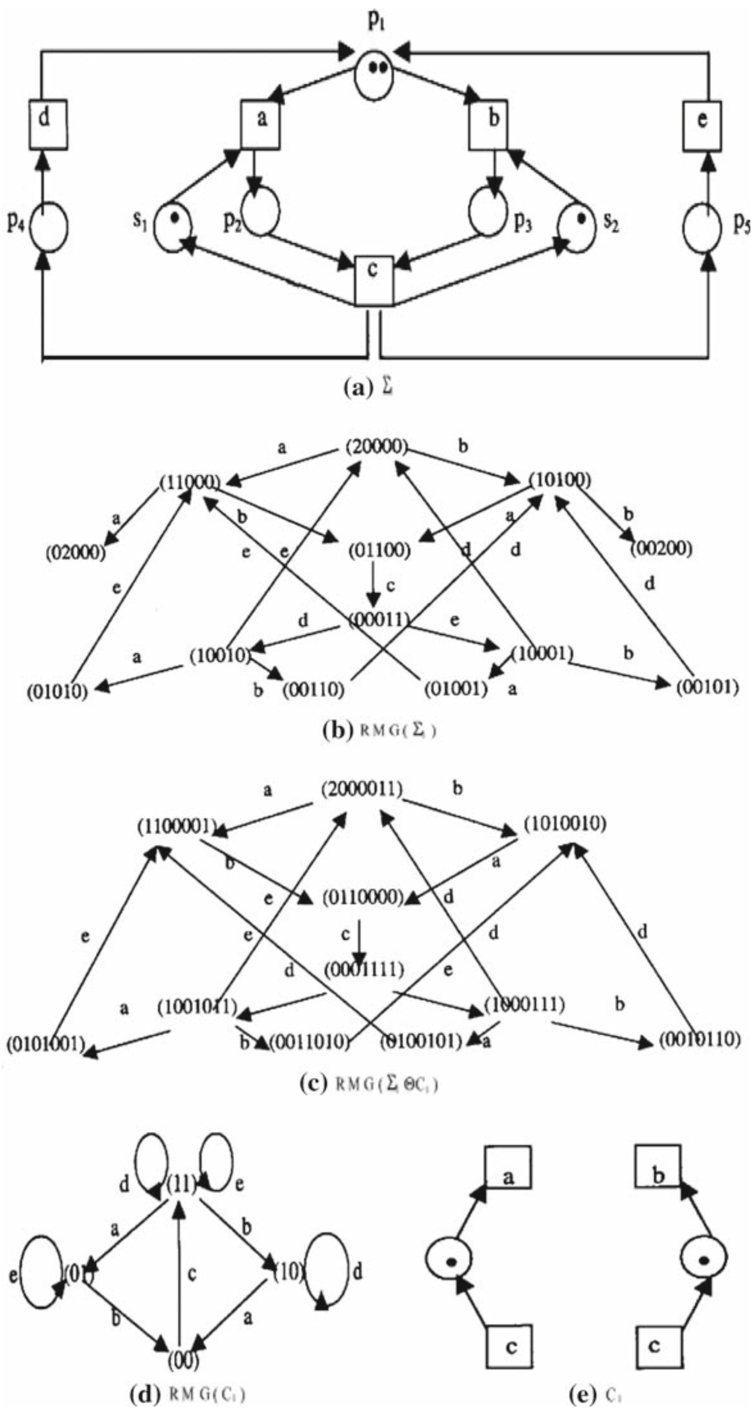

Fig. 3 Synthetical picture of deadlock controller

Step 5. Heap stack, go to Step 2. According to the state equation of the network, we can know:

$$M_c - \text{stack(top)} = A^T \mathbf{t}$$

Here A^T is the transposed matrix of connection matrix of net C, t = f((stack(top),M_cc)). In this way, we can get the structure (P_c,T_c,F_c) of net C from every two nodes of RMG(C) and correlative edges. Then according to the state equation mentioned before, we can get initial label Mc_0 of C from label of initial node of RMG(C), thereby we get net C = (P_c,T_c,F_c, Mc_0).

6 Dealing with dead lock phenomenon

Dead lock phenomenon is an abnormal phenomenon which takes place easily in concurrent system. If it is not eliminated, it will lead to the paralysis of the whole system. Now, we discuss the problem of dead lock control. $\Sigma 1$ is a Petri net in Fig. 3a, b is reachable state fig. RMG($\Sigma 1$) of $\Sigma 1$. It is easy to see that there are two dead lock states: (02000) and (00200). In order to eliminate the dead lock states we bring in controlling place s1 and s2, and set control criterion Spce = {$M(p_2) < 2$, $M(p_3) < 2$, $M(p_2) + M(s1) = 1$, $M(p_3) + M(s2) = 1$ |$M \in R(Mo)$}.

For $Mo(p_2) = Mo(p_3) = 0$, so $Mo(s1) = Mo(s2) = 1$. Change RMG($\Sigma 1$) into RMG ($\Sigma 1 \Theta$ C1) (see Fig. 3c). We can get RMG(C1) (see Fig. 3d) and C1 (see Fig. 3e) according to algorithm 1 and 2, respectively. C1 that we get in his way is Petri net model of controller. We must notice that the structure of C1 may not be unique.

7 Summary

This paper has proposed concept of fuzzy Petri net to be against limitation of the classical Petri net and also give a method of controller design based on Fuzzy Petri net. It points out that this is an algorithm to maximum feedback allowed controller of Fuzzy Petri net. The controller is composed of places and arcs. Its size is in proportion to the number of constraints. This algorithm can be put into practice easily because it only involves the vector operation and the matrix operation. We can utilize this method to solve dead lock phenomenon in concurrent system. Then, our research can sum up the controller design of Fuzzy Petri net as a mathematical problem (Zhong 2005; Zhong 2006; Zhong 1994), which is a matrix equation or a matrix inequality problem. Does the solution to mathematical problem exist doubtless? What are the conditions assure it of existing? How to get the special solution if it exists? If there exist many solutions, we can raise the design problem of the most optimized controller (Miao and Li 2004; Li 1997) under some certain condition.

Acknowledgements This research is supported by the National Natural Science Foundation of China (No. 70771030).

References

Chen, Z., & Zhong, Y. (2003). Design of adaptive inserting rule fuzzy controllers. *Chinese Control Theory & Applications, 20*(1), 129–132.

Holloway, L. E., & Krogh, B. H. (1990). Synthesis of feedback logic for a class of control Petri nets. *IEEE Transactions On Automatic Control, 35*(5), 514–520.

Li, H. (1997). The essence of Fuzzy control and a kind of fine Fuzzy controller. *Control Theory & Applications, 14*(6), 868–872.

Looney, C. (1988). Fuzzy Petri nets for Rule_based decisionmaking. *IEEE Transactions on SMC, 18*(1), 638–646.

Miao, Z., & Li, H. (2004). Design of control system with robust pole placement constraints. *Control Theory & Applications, 21*(5), 729–734.

Wang, Q., & Zhu J. (1998). A synthetical method of feedback control for controlled Petri net. *Information and Control, 27*(5), 326–330.

Zhong, Y., (1994). Imitating and forecast of economic systems of district industry of central district of city. *Chinese Journal of Decision Making and Decision Support Systems, 4*(4), 91–97.

Zhong, Y. (2005). The FHSE model of software system for synthetic evaluating enterprising. *Journal of Guangzhou University, 4*(4), 316–320.

Zhong, Y. (2006). The researching and realizing in ERP system based on C/S structure mode mixed with B/S. *Chinese Journal of Information of Microcomputer, 11*, 31–33.

Zhou, M. C., Dicesare, F., & Desrochers, A. A. (1992). A hybrid methodology for symthesis of petri net models for manufacturing system. *IEEE Transactions Robotics & Automation, 8*(3), 356–361.