Article ID: 1000 - 8152(2003)02 - 0228 - 05

Synthesis of Petri nets controller for discrete event systems based on finite capacity places - Part 1

WU Wei-min, SU Hong-ye, CHU Jian

(National Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Zhejiang Hangzhou 310027, China)

Abstract: A novel method of controller design for discrete event systems (DES) modeled by Petri nets (PN) is presented. The controller is constructed based on the concept of finite capacity places such that the plant evolves under the constraint of linear inequalities defined on the place marking. The synthesis procedure of the controller exploits the transformation technique of finite capacity Petri net to an infinite one. PN controller synthesis algorithms are presented for different cases of the constraint.

Key words: discrete event systems; controller synthesis; Petri nets; finite capacity places

CLC number: O158 **Document code**: A

基于有限容量库所的离散事件系统的 Petri 网控制器综合——第 1 部分

吴维敏，苏宏业，褚　健

(工业控制技术国家重点实验室 浙江大学 先进控制研究所，浙江 杭州 310027)

摘要：针对由 Petri 网建模的离散事件系统，提出了一种新的控制器设计方法.控制器是基于有限容量库所的概念构造而成的，并使被控对象在给定的一组线性不等式约束下运行，而给定的线性不等式约束是定义在库所标识上的.控制器的综合利用了有限容量库所 Petri 网转换为(普通)无限容量库所 Petri 网的技术.针对约束的不同情况，给出了相应的 Petri 网的控制器的设计方法.

关键词：离散事件系统；控制器综合；Petri 网；有限容量库所

1 Introduction

Automata and Petri nets are the main modeling tools in the research area of control synthesis for discrete event systems. Petri net models are examined in the DES control synthesis by many researchers due to the advantage of the graphical and distributed representation of the system state and the computational efficiencies. In this paper we propose a new method for synthesizing the controller of DES modeled by PN. The control objective is to force the plant to satisfy the logical conjunction of separate linear inequality constraints, which have the following form

$$\sum_{i=1}^{n} l_i \mu_i \leqslant b, \tag{1}$$

where μ_i is the marking of place p_i, b is an integer constant, n is the place number of the net, while coefficient l_i may be any integer. Though it has already been addressed by Yamalidou et al[1], the constraint will be studied from a new standpoint in this paper.

Many other constraints specification can be categorized into this kind of constraint. Forbidden state (or state avoidance) problem, which was surveyed from the time when the control theory of DES was initiated[2], can be expressed as linear inequality constraint in some case. In [2], Ramadge and Wonham proposed a state feedback solution based on automata. In order to avoid the numerous state representations of automata, the subsequent researchers use Petri nets to study the control method of this problem. Holloway and Krogh finished the early work of this aspect in [3 ~ 5]. Another con-

Received date: 2001 - 08 - 02; Revised date: 2002 - 06 - 24.

Foundation item: supported by the National Natural Science Foundations of China (60025308); Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE, P.R.C; the Doctor Degree Program Foundation of China (20020335103).

straint specification is so-called generalized mutual exclusion constraint (GMEC) in the framework of Petri nets[6]. GMEC is a special case of linear inequality constraint discussed in this paper when the coefficient l_i is confined to nonnegative integer. The specification of GMEC can be enforced by a set of places called monitors in [6].

The new method for synthesizing Petri nets controller introduced in this paper is based on the concept of finite capacity places[7, 8] and is named FCP method. The inequality (1) can be regarded as the capacity constraint forced upon the related places, and consequently the plant can be treated as a finite capacity Petri nets. Thus the transformation technique from finite capacity Petri nets to an infinite one can be utilized to design the controller.

2 Finite capacity Petri nets

Usually, it is assumed that each place in a Petri net can accommodate an unlimited number of tokens. To model many physical systems, it is natural to consider an upper limit to the number of tokens that each place can hold. Such a Petri net is named finite Petri net[7,8].

Definition 1 In a finite capacity Petri net, each place p has an associated capacity Cap (p), the maximum number of tokens that p can hold at any time.

An enabled input transition of a place p with capacity Cap (p) has the possibility of firing when the firing of this transition does not result in a number of tokens in p which exceeds this capacity.

Using the technique of complementary place, the transformation of a finite capacity Petri net into an infinite one is rather simple. For the detailed transformation procedure, the reader may refer to [7, 8].

In the following, we will describe how finite capacity places can be used to synthesize a DES controller.

3 Controller synthesis using finite capacity places

The plant discussed in this paper may be generalized Petri net or just an ordinary one. For simplicity, we firstly consider the ordinary Petri nets. What is more, in this paper we restrict the coefficient l_i in (1) to a Boolean variable and the constraint contains the place p_i when the value is true. However the synthesis method given in this paper can also be applied to the general case where the plant is modeled by generalized Petri nets and the constraint efficient is not confined to Boolean variable.

3.1 The Simplest case

When the coefficient l_i in (1) is a Boolean variable, this case was studied as unweighted GMEC in [6] and was referred as set condition in [4]. Assume $n = 3$, $b = 2$ and $l_i (i = 1,2,3)$ is true in the Ineq. (1), the constraint means that the total number of the tokens in the places p_1, p_2 and p_3 cannot exceed 2.

Now let us consider a more special case. Assume only one coefficient in the constraint (1), for instance, l_3 is true, then the inequality has the form below:

$$\mu_3 \leqslant b. \tag{2}$$

We notice that constraint (2) has the exact mean as the finite capacity Petri nets. The maximum number that place p_3 can hold at any time is b. This is also the control objective that the controller wants to force the plant to obey. we can utilize the transformation technique of finite capacity Petri nets to design the controller.

For the convenience of description, we make the following definitions before presenting the detailed steps of the controller design.

Definition 2 The place in the constraint Ineq. (1) is named constrained place. The entire constrained places constitute constrained place set, denoted by C_p, that is

$$C_p = \{p = p_i \mid \sum_{i=1}^{n} l_i \mu_i \leqslant b \text{ for } l_i \neq 0\}. \tag{3}$$

Definition 3 The set of input transitions for the entire constrained place set C_p is said to be input constrained transition set, denoted by ${}^{\circ}C_p$, that is

$${}^{\circ}C_p = \{t \mid t \in {}^{\circ}p \text{ for } p \in C_p\}, \tag{4}$$

where the notation ${}^{\circ}p$ means the input transitions of p.

Similarly, we have the definition of output constrained transition set:

Definition 4 The set of output transitions for the entire constrained place set C_p is said to be output constrained transition set, denoted by $C_p{}^{\circ}$, that is

$$C_p{}^{\circ} = \{t \mid t \in p^{\circ} \text{ for } p \in C_p\}, \tag{5}$$

where the notation p° means the output transitions of p.

Definition 5 The set of transitions denoted by CC_t

is said to be common constrained transition set, if its element t satisfies

$$t \in {}^{\circ}C_p \cap C_p{}^{\circ}, \tag{6}$$

that is, some input places and output ones of the transition in the CC_t are all constrained places.

Definition 6 Given the input constrained transition set ${}^{\circ}C_p$, the set ${}^{\circ}C_{\text{pure}-t} = {}^{\circ}C_p - CC_t$ is said to be pure input constrained transition set.

Definition 7 Given the output constrained transition set $C_p{}^{\circ}$, the set $C^{\circ}{}_{\text{pure}-t} = C^{\circ}{}_p - CC_t$ is said to be pure output constrained transition set.

The constraint of the type (1) can be regarded as the enforcement of finite capacity on all the places in the set C_p. Then the plant can be regarded as a finite capacity Petri net. The sum of tokens in each place in the set C_p cannot exceed b at any time. We can regard all the elements in C_p as one place when we synthesize the desired controller that satisfies the constraint (1).

With the above definitions, the design method of the controller conforming to the constraint (1) is summarized as the following Algorithm 1.

Algorithm 1

1) For the constraint (1), evaluate the constrained place set C_p, the input constrained transition set ${}^{\circ}C_p$, the output constrained transition set $C_p{}^{\circ}$, the common constrained transition set CC_t, the pure input constrained transition set ${}^{\circ}C_{\text{pure-}t}$, and the pure output constrained transition set $C^{\circ}{}_{\text{pure-}t}$.

2) For each $t \in {}^{\circ}C_{\text{pure-}t}$, draw an arc from the controller place p_c to this transition, that is, let p_c be an input place of the transition t.

3) For each $t \in C^{\circ}{}_{\text{pure-}t}$, draw an arc from this transition to the controller place p_c, that is, let p_c be an output place of the transition t.

4) According to the following Eq. (7), calculate the initial marking of p_c, that is

$$M_0(p_c) = b - \sum_{i=1}^{n} l_i \mu_{i0}, \tag{7}$$

and the following equation is always satisfied

$$M(p_c) = b - \sum_{i=1}^{n} l_i \mu_i. \tag{8}$$

This can be deduced from the connective relation (Step 2 and Step 3) between controller and the plant. If the constrained places get (lose) tokens, the controller is sure to lose (get) the same amount of tokens.

The controller designed using above method enforces the plant to obey the given constraint (1). The transition $t \in {}^{\circ}C_{\text{pure-}t}$ can enable and fire if and only if the controller is marked according to the enable rule of transitions. The fact that the controller is marked implies the constraint inequality holding, which can be deduced from Eq. (8) directly. From (8), there is

$$\sum_{i=1}^{n} l_i \mu_i = b - M(p_c), \tag{9}$$

and because $M(p_c) \geqslant 0$, there is

$$\sum_{i=1}^{n} l_i \mu_i \leqslant b. \tag{10}$$

When the enabling transition $t \in {}^{\circ}C_{\text{pure-}t}$ fires, the total token number of constrained places increases while the controller lose the same amount of tokens. When the Ineq. (1) acquires the maximum b, the controller has no token and the transitions in ${}^{\circ}C_{\text{pure-}t}$ are disabled. This is the role that the controller should play. With the firing of the transitions in $C^{\circ}{}_{\text{pure-}t}$, the controller gets tokens and there is again possibility for the firing of the transitions in ${}^{\circ}C_{\text{pure-}t}$.

In Algorithm 1, the case when the constrained places have common input and/or common output transitions is not considered. This case will be discussed in [9]. Without making any special specifications, in this paper we assume there are no common input and/or common output transitions of constrained places.

As an example consider the Petri net of Fig. 1[1], which is acyclic and nonsafe. Its initial marking is $\mu_{p0} = [\mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4]^{\mathrm{T}} = [3\,0\,0\,3]^{\mathrm{T}}$. Our control objective is to ensure that the places p_2 and p_3 never contain more than one token at any time, i.e., the constraint should satisfy

$$\mu_2 + \mu_3 \leqslant 1. \tag{11}$$

We regard the place p_2 and p_3 as one place and its maximum capacity is 1. Thus the plant may be treated as a finite capacity Petri net. First, we evaluate various sets needed in the design procedure. In this example, we have $C_p = \{p_2, p_3\}$, ${}^{\circ}C_p = \{t_1, t_2, t_3, t_5\}$, $C_p{}^{\circ} = \{t_2, t_3, t_4\}$, $CC_t = \{t_2, t_3\}$, ${}^{\circ}C_{\text{pure-}t} = \{t_1, t_5\}$ and $C^{\circ}{}_{\text{pure-}t} = \{t_4\}$. Then we draw arcs between the transitions in

${}^{\circ}C_{\text{pure-}t}$ and $C^{\circ}{}_{\text{pure-}t}$ and the controller place p_c. The transitions in ${}^{\circ}C_{\text{pure-}t}$ are the end of the arcs and the one in $C^{\circ}{}_{\text{pure-}t}$ is the beginning of the arcs. By (7), the initial marking of p_c is calculated as follows:

$$\mu_{pc} = b - \sum_{i=1}^{n} l_i\mu_i = 1.$$

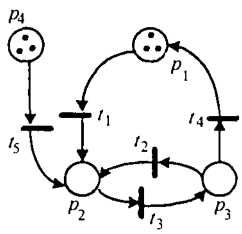

Fig. 1 Plant graph for the example of Section 3.1

The plant with the addition of a controller p_c is shown in Fig. 2. The arcs between the plant and p_c are illustrated with dashed lines, and the controller place itself is also distinguished from the places in the plant by thicker circle.

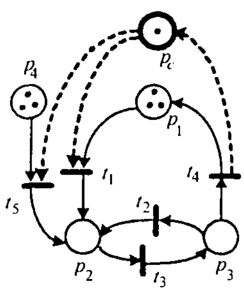

Fig. 2 The plant of Fig.1 with the addition of a controller

We have the same controller as the one in [1], but our method need not consider the entire plant net, because the incidence matrix and its operation are not used here. What we deal with in this method is only the places and its input and output transitions related to the constraints. Thus, our method is simpler and more efficient in computation. When the plant is a large system, the advantage of this design method is more obvious, which will be illustrated in the example in [9].

3.2 The case of 'greater than or equal to' constraint

This case is corresponding to the one studied in Section 3.1 and is referred to as

$$\sum_{i=1}^{n} l_i\mu_i \geqslant b. \tag{12}$$

Note that l_i is still a Boolean variable, and the other notations have the same meaning as that in the Ineq.(1).

Similarly, the constraint of the type (12) can still be regarded as the enforcement of finite capacity on all the places in the set C_p. But the meaning of 'finite capacity' is different from that in Section 3.1, where the maximum capacity that the places in the set C_p can hold. What is indicated here is that the minimum capacity that the entire constrained places should hold in the evolution of the plant at any time. Consequently, the plant can still be regarded as a finite capacity Petri nets. The sum of tokens in each place in the set C_p must exceed or equal to b at any time. What should we do is synthesize the expected controller to achieve the objective.

The design steps of the controller with given constraint (12) are summarized as the following Algorithm.

Algorithm 2

1) This step is the same as the one in Algorithm 1; that is, for the constraint (12), evaluate the sets C_p, ${}^{\circ}C_p$, $C_p{}^{\circ}$, CC_t, ${}^{\circ}C_{\text{pure-}t}$ and $C^{\circ}{}_{\text{pure-}t}$.

2) For each $t \in {}^{\circ}C_{\text{pure-}t}$, draw an arc from this transition to the controller place p_c; that is, let p_c be an output place of the transition t.

3) For each $t \in C^{\circ}{}_{\text{pure-}t}$, draw an arc from the controller place p_c to this transition; that is, let p_c be an input place of the transition t.

4) According to the following equation, calculate the initial marking of p_c, that is

$$M_0(p_c) = \sum_{i=1}^{n} l_i\mu_{i0} - b, \tag{13}$$

and the following equation is always satisfied

$$M(p_c) = \sum_{i=1}^{n} l_i\mu_i - b. \tag{14}$$

By (14), we know the controller enforces the plant to obey the given constraint (12). From (14),

$$\sum_{i=1}^{n} l_i\mu_i = b + M(p_c). \tag{15}$$

In addition, $M(p_c) \geqslant 0$, thus the following equation holds

$$\sum_{i=1}^{n} l_i\mu_i \geqslant b. \tag{16}$$

This is the Ineq. (12). When the total token number of constrained places acquires the minimum b, the controller has no token and the transitions in $C^{\circ}{}_{\text{pure-}t}$ are disabled.

Consider the Petri net of Fig. 1 again, but its initial marking is $\mu_{p0} = [\mu_1 \ \ \mu_2 \ \ \mu_3 \ \ \mu_4]^{\mathrm{T}} =$

$[3\ \ 0\ \ 1\ \ 3]^{\mathrm{T}}$ now. For clarity, we draw the Petri net of Fig. 1 in Fig. 3 again. The control goal is to ensure that there always exists at least one token in the places p_2 and p_3 at any time, i.e. the constraint should satisfy

$$\mu_2 + \mu_3 \geqslant 1. \tag{17}$$

We regard the places p_2 and p_3 as one place and its minimum capacity is 1.

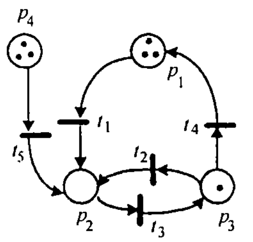

Fig. 3 Plant graph for the example of the Section 3.2

By Algorithm 2, the controlled system shown in Fig.4 can be obtained.

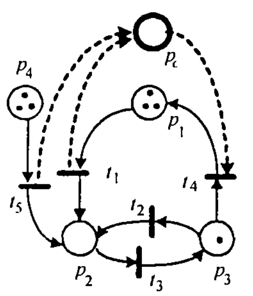

Fig. 4 The plant of Fig.3 with the addition of a controller

4 Conclusions

In this paper we present a new method named FCP method for synthesizing Petri nets controller for discrete event systems. The basic idea behind FCP method is to regard the constraint as places with finite capacity. The desired controller can be obtained by adding a complementary place of the constrained places. This method is stimulated by the transformation technique from finite capacity Petri nets to an infinite one. But their connections between the original net (plant) and the complementary place (controller place) are different since the constrained places are usually the linear combination of places and not a single place. The principle of the connection between the plant and the controller place prevents the sum of tokens in constrained places from exceeding the given number.

In the sequel paper [9], the general case of the constraint shall be considered and the maximal permissiveness of FCP method shall be proved. In addition, we shall consider the example of AGV (automated guided vehicles) coordination system in the literature to illustrate the advantages and characteristics of FCP method.

References:

[1] YAMALIDOU K, MOODY J O, LEMMON M D, et al. Feedback control of Petri nets based on place invariants [J]. *Automatica*, 1996,32(1):15 - 28.

[2] RAMAGE P J, WONHAM W M. Modular feedback logic for discrete event systems [J]. *SIAM J Control Optimization*, 1987,25(5):1202 - 1218.

[3] HOLLOWAY L E, KROGH B H. Synthesis of feedback logic for a class of controlled Petri nets [J]. *IEEE Trans on Automatic Control*, 1990, 35(5): 514 - 523.

[4] KROGH B H, HOLLOWAY L E. Synthesis of feedback logic for discrete manufacturing systems [J]. *Automatica*, 1991,27(4):641 - 651.

[5] HOLLOWAY L E, GUAN X, ZHANG L. A generalization of state avoidance policies for controlled Petri nets [J]. *IEEE Trans on Automatic Control*, 1996,41(6):804 - 816.

[6] GIUA A, DICESARE F, SILVA M. Generalized mutual exclusion constraints on nets with uncontrollable transitions [A]. *Proc of 1992 IEEE Int Conf on Systems, Man, and Cybernetics* [C]. Chicago, IL: Institute of Electrical & Electronics Engineers, Inc, 1992, 974 - 979.

[7] DAVID R, ALLA H. *Du Grafcet Aux Réseaux de Petri* [M]. Paris: Hermès, 1989.

[8] MURATA T. Petri nets: properties, analysis, and applications [J]. *Proc of the IEEE*, 1989,77(4):541 - 580.

[9] WU Weimin, SU Hongye, CHU Jian. Synthesis of petri nets controller for discrete event systems based on finite capacity places - Part 2 [J]. *Control Theory & Applications*, 2003,20(2):233 - 238.
(吴维敏,苏宏业,褚健.基于有限容量库所的离散事件系统的Petri网控制器综合——第2部分[J].控制理论与应用,2003,20(2):233 - 238.)

作者简介:

吴维敏 (1970 —),男,1996年毕业于太原重型机械学院,1999年2月和2002年3月分别获太原重型机械学院硕士学位和浙江大学博士学位,现为浙江大学讲师,研究兴趣为离散事件系统和混杂系统,E-mail:wmwu@iipc.zju.edu.cn;

苏宏业 (1969 —),男,1990年毕业于南京化工大学,1995年获浙江大学工业自动化专业博士学位,现为浙江大学先进控制研究所副所长,教授,博士生导师,主要研究兴趣是鲁棒控制,时滞系统控制,非线性系统控制和PID自整定理论和应用研究;

褚 健 (1963 —),男,1982年毕业于浙江大学,1989年获工学博士学位,1993年聘为浙江大学教授,博士生导师,现为工业自动化国家工程研究中心副主任,工业控制技术国家重点实验室主任,首批“长江”学者计划特聘教授,浙江大学先进控制研究所所长,主要从事时滞系统,控制,非线性控制,鲁棒控制等理论与应用研究.