Reducing the Number of Constraints in Non Safe Petri Net

M. Zareiee, A. Dideban

***Abstract*—**This paper addresses the problem of forbidden states in non safe Petri Nets. In the system, for preventing it from entering the forbidden states, some linear constraints can be assigned to them. Then these constraints can be enforced on the system using control places. But when the number of constraints in the system is large, a large number of control places must be added to the model of system. This concept complicates the model of system. There are some methods for reducing the number of constraints in safe Petri Nets. But there is no a systematic method for non safe Petri Nets. In this paper we propose a method for reducing the number of constraints in non safe Petri Nets which is based on solving an integer linear programming problem.

***Keywords*—**discrete event system, Supervisory control, Petri Net, Constraint

I. Introduction

SUPERVISORY control theory which was presented by ramadge and wonham is a general theory for controlling the behavior of discrete event systems (DES) [1], [2]. This theory tries to verify a given specification by restricting the behavior of the system. This restriction can be performed by disabling some controllable events [3].

Automata is a tool for modeling discrete event systems which was used by the pioneers of the supervisory control theory. Automata is a state-transition tool and when the number of states is too large, modeling systems by this tool is difficult or maybe impossible [4]. So, Petri net (PN) has been proposed as an alternative tool for modeling these systems [5]. Compact structure and mathematical properties have made PN as a useful tool for modeling discrete event systems. In PNs, each event is assigned to a transition and the control is performed on the transitions by controllable events (controllable transitions). Existing uncontrollable events in the system may lead to entering the system in the forbidden states. So the system must be prevented from entering the forbidden states. But by disabling controllable events in special conditions, preventing system from entering the forbidden states is possible.

For preventing the system from entering the forbidden states some efforts has been accomplished. In [6] the authors proposed to put some conditions on the controllable transitions. These conditions are calculated online and lock the controllable transitions in special cases. A similar method is presented in [7] by this difference that the conditions are obtained from the marking graph and are calculated offline. Another method for preventing the system from entering the forbidden states is adding some places to the PN model of system. These places which are called control places restrict the behavior of the system for obtaining the objective function. In [8] a method for calculation of control places is offered. This method by knowing the forbidden states adds some control places to the system to forbid the forbidden states. Another method for adding the control places has been proposed in [9]. This method adds a control place instead of each constraint.

In [10] a method is proposed for assigning constraint to forbidden states in safe PNs. then the constructed constraints can be enforced on the system using the idea in [9]. But when the number of forbidden states and consequently the number of constraints is large, adding control places to the system makes the system complicated. In [10], it is shown that the number of constraints can be reduced. It means that it is possible to use a constraint instead of some constraints without forbidding any authorized states.

In [11] a method for reducing the number of constraints in safe PN has been proposed. This method uses the invariant and partial invariant properties for this reduction. Another method for simplification of constraints in safe PN has been proposed in [12] which using the over-state concept reduces the number of constraints. In [13] another method is proposed that like the last method uses the over-state concept for reducing the number of constraints. This method performs the reduction by using the relation between the over-states of forbidden states and authorized states. But all of the mentioned methods are applicable on safe PNs and we cannot apply them on non safe PN.

In this paper we propose an algorithm for reducing the number of constraints in non safe Petri Nets by considering a generic constraint. In this algorithm, we want to find the constant coefficients in the generic constraint to consider it as a constraints verifying all the constraints related to forbidden states. For finding these constants, we solve an integer linear programming problem (ILP) which is composed of some inequalities that some of them verify the authorized states and some of them violate the forbidden states. When this algorithm has an answer, we can reduce the number of constraints to one constraint.

The rest of this paper is organized as follows. In section II, the basic concepts which are necessary for presenting the

M. Zareiee is with the Semnan University, Semnan, IRAN, (e-mail: meisamzareiee@ Gmail).

A. Dideban, is with Semnan University, Semnan, IRAN. He is with Semnan University, Semnan, IRAN (corresponding author to provide phone:(+98)231-3354123; fax: (+98)231-3354123; e-mail:a_dideban@yahoo.com).

paper are considered. The new method for reducing the number of constraints is proposed in section III. Finally conclusion is presented in section IV.

II. Preliminary presentation

A. Place/Transition net

A PN is represented by a quadruplet $R=\{P, T, W, M_0\}$ where P is the set of places represented by circles, T is the set of transitions represented by bars. $P \cap T \neq \phi, P \cup T \neq \phi$. W is the incidence matrix and M_0 is the initial marking. A marking is a $m \times 1$ vector where m is the number of places and assigns to each place of a Place/Transition net a non-negative integer number of tokens. Places and transitions are connected together by arcs.

PNs are divided to two subsets: safe PN and non safe PN. Safe PNs are the ones that the number of tokens in its places cannot be more than one. But in non safe PNs this number can be more than one.

In an industrial system, the model of system can be divided into two sections. The first section is the model of process and the next is the model of specification. The process model of system is the model of components of system and the specification is the model of some conditions that must be verified by the system for obtaining the desired behavior.

In a PN model of a system, all of the states which can be obtained by the model compose the set of reachable states and this set is shown by $\mathcal{M}_R$. In the set $\mathcal{M}_R$ there may be some states that violate specifications or are deadlock states or the ones that the occurrence of uncontrollable events leads to these states. These states are called forbidden states [12]. So, the set $\mathcal{M}_R$ can be divided to two subsets. The first one is the set of forbidden states and is shown by $\mathcal{M}_F$ and the other one is the set of authorized states and is shown by $\mathcal{M}_A$. These two subsets don't have any common component.

In the set of forbidden states, there is a very important subset that is called the set of border forbidden states [14]. Forbidding these states leads to forbidding all the forbidden states. These states are defined as follows:

Definition 1 ([12]): Let $\mathcal{M}_B$ be the set of border forbidden state:

$$\mathcal{M}_B = \{M_i \in \mathcal{M}_F \mid \exists \sigma \in \textstyle\sum_c \, and \exists M_j \in \mathcal{M}_A, M_j \xrightarrow{\sigma} M_i\}$$

Where $\sum_c$ is the set of controllable transitions. ❑

From definition 1, it is obvious that by disabling controllable events when the firings of them are leading to the border forbidden states, preventing system from reaching to the border forbidden states is possible and then the system cannot reach to any forbidden states.

B. control places

To calculate the control places corresponding to each linear constraint, the method introduced in [9] is used. This method is based on the concept of invariant, and now it is briefly introduced. Consider the set of constraints as $L.M_P \leq b$ where M_P is the marking vector, L is a $n \times n$ matrix, b is a $n_c \times 1$ vector, n_c is the number of constraints and n is the number of places. In this method for each constraint, a place is added to the model. Let W_p be the PN incidence matrix. For each constraint a row is added to W_p and these rows are shown as W_c, calculated as follows:

$$W_c = -L.W_p$$

W_c is added to W_p as follows:

$$W = \begin{bmatrix} W_p \\ W_c \end{bmatrix}$$

If the initial marking of the model is M_{P0}, the initial marking for the added places is:

$$M_{s0} = b - L.M_{P0}$$

The final initial marking of the controlled model is:

$$M_0 = \begin{bmatrix} M_{p0} \\ M_{s0} \end{bmatrix}$$

However when the number of control places is large, the controller (PN model of system) is complicated. This leads to the necessity to reduce the number of constraints [10].

III. Reducing the number of constraints

In a system when the number of constraints is large, a large number of control places must be added to the PN model of system. But in [10] it is shown that the number of constraints can be reduced. It means that it is possible to apply a constraint which verifies some constraints. So, very active research in the field of reducing the number of constraints emerged during the recent years. In [11] a method is proposed that reduces the number constraints by the invariant and partial invariant properties. This method is applicable on safe and conservative PNs. Another method for reducing the number of constraints is proposed in [12]. This method is more general than the first method and is applicable in safe PN. This method uses the over-state concept and chooses the over-states which forbidding them lead to the forbidding all of the forbidden states and verifying all of the authorized states. Then by a method like the McCluskey method for simplifying logical expressions, chooses the smallest number of over-states which forbidding them lead to forbidding all of the forbidden states. Another method for reducing the number of constraints is proposed in [13] which develops the idea in [12]. This method uses the concept of over-state too and can reduce the number of resultant constraints by the method in [12]. But all of the mentioned methods are applicable on safe PN. Therefore, we want to introduce a method which can be applicable on safe and non safe PNs. in the next section we discuss this problem.

A. Reducing the number of constraints by considering a generic constraint

In this section we want to introduce a method to reduce the number of constraints in safe and non safe PNs. For this reason we can consider a generic constraint as follows:

$$k_1m_1+k_2m_2+\ldots+k_nm_n\leq x \quad (1)$$

where x and k_i for i=1,…, n are constants and n is the number of places of PN model and m_i is the number of tokens in place P_i. if we can obtain x and k_i for i=1,…, n as the mentioned inequality verify all of the authorized states and forbid all of the forbidden states, then we can consider the resultant inequality as a constraint for preventing the system from entering all of the forbidden states. Verifying the forbidden states by this inequality are obtained when the term ($k_1m_1+k_2m_2+\ldots+k_nm_n$) is smaller than or equal to x for all of the authorized states and violating the forbidden states by this inequality is obtained when the term ($k_1m_1+k_2m_2+\ldots+k_nm_n$) is greater than x for all of the forbidden states. So, to verify the authorized states by this inequality, we put all of the authorized states in the inequality (1) and obtain a set of inequalities. Then for violating the forbidden states by the inequality (1), we put all of the forbidden states in the inequality (1) and convert the smaller equal sign to greater sign to obtain another set of inequalities. Now we consider the two sets and solve them to obtain an answer. This is an integer linear programming problem where the objective function is: minimum ($k_1+k_2+\ldots+k_n+x$) in which x>0 and k_i≥0 for i=1,…n. when this algorithm has an answer we can consider a constraints for forbidding a group of forbidden states. This concept reduces the number of control places which must be added to the system. Now we generate this concept in algorithm 1.

Algorithm 1. Let $\mathcal{M}_A=\{P_{A11}^{s_{A11}}P_{A12}^{s_{A12}}\ldots P_{A1t}^{s_{A1t}},\ldots,P_{Ar1}^{s_{Ar1}}P_{Ar2}^{s_{Ar2}}\ldots P_{Arq}^{s_{Arq}}\}$ be the set of authorized states and $\mathcal{M}_B$={ $P_{B11}^{s_{B11}}P_{B12}^{s_{B12}}\ldots P_{B1u}^{s_{B1u}},\ldots,P_{Bg1}^{s_{Bg1}}P_{Bg2}^{s_{Bg2}}\ldots P_{Bgh}^{s_{Bgh}}$, } the set of forbidden states. Follow these steps to obtain a constraint related to these forbidden states:

Step 1: Consider an inequality as follows:

$$k_1m_1+k_2m_2+\ldots+k_nm_n\leq x \quad (2)$$

Where n is the number of places and m_i is the number of tokens in place P_i, and x and k_i for i=1,2,…,n are constants.

Step 2: Put the markings of all the authorized states in the inequality (2) and construct inequalities as follows:

$$P_{A11}^{s_{A11}}P_{A12}^{s_{A12}}\ldots P_{A1t}^{s_{A1t}}\rightarrow k_{A11}s_{A11}+k_{A12}s_{A12}+\ldots+k_{A1t}s_{A1t}\leq x \quad (1-3)$$

.

.

.

$$P_{Ar1}^{s_{r1}}P_{Ar2}^{s_{Ar2}}\ldots P_{Arq}^{s_{Arq}}\rightarrow k_{Ar1}s_{Ar1}+k_{Ar2}s_{Ar2}+\ldots+k_{Arq}s_{Arq}\leq x \quad (f-3)$$

Where f is the number of authorized states.

Step 3: Put the marking of all the border forbidden states in the inequality (2) and convert the smaller equal sign to greater sign as follows:

$$P_{B11}^{s_{B11}}P_{B12}^{s_{B12}}\ldots P_{B1u}^{s_{B1u}}\rightarrow k_{B11}s_{B11}+k_{B12}s_{B12}+\ldots+k_{B1u}s_{B1u}>x \quad (1-4)$$

.

.

.

$$P_{Bg1}^{s_{Bg1}}P_{Bg2}^{s_{Bg2}}\ldots P_{Bgh}^{s_{Bgh}}\rightarrow k_{Bg1}s_{Bg1}+k_{Bg2}s_{Bg2}+\ldots+k_{Bgh}s_{Bgh}>x \quad (v-4)$$

Where v is the number of border forbidden states.

Step 4: Solve the set of relations (1-3) to (f-3) and (1-4) to (v-4) which is an integer linear programming (ILP) problem and obtain the minimum values of x and k_i for i=1,2,…,n. (in this problem the objective function is: minimum($k_1+k_2+\ldots+k_n+x$) where x>0 and $k_i \geq 0$ for i=1,2,…,n)

Step 5: If step 4 has an answer, then put x and k_i for i = 1,2,…,n in the inequality (2). The resultant inequality is a constraint for the forbidden states $P_{B11}^{s_{B11}}P_{B12}^{s_{B12}}\ldots P_{B1u}^{s_{B1u}},\ldots,P_{Bg1}^{s_{Bg1}}P_{Bg2}^{s_{Bg2}}\ldots P_{Bgh}^{s_{Bgh}}$.

❑

Now, we want to see this concept on the simple example.

Example 1: in this example we consider a net which is represented in figure 1. In this example the transition t_1 and t_2 are controllable and the transition t_3, t_4 and t_5 are uncontrollable.

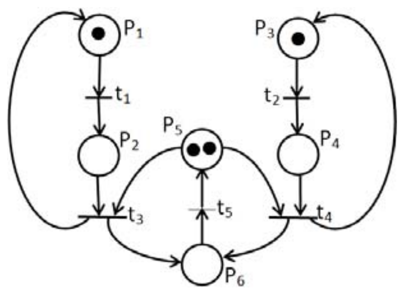

Fig. 1. Nt system used in example 1

Marking graph of this system is shown in figure 2. In this figure, the uncertain states are shown by ⨂. These states are those which violate specification. For example, when the system is in the state $P_1P_4P_6^2$ firing of the transition t_4 violates the specification.

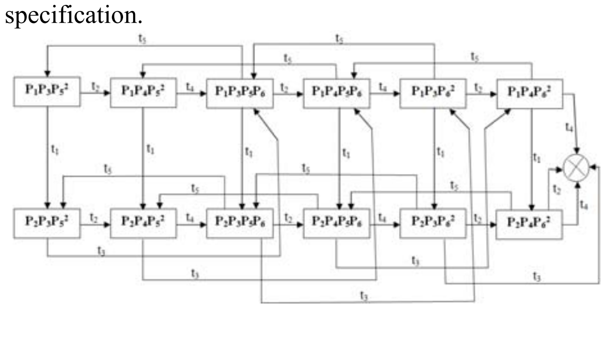

Fig. 2. Marking graph of the system in example 1

Looking at the marking graph, the set of authorized states is:

$\mathcal{M}_A=\{P_1P_3P_5^2,\ P_1P_4P_5^2,\ P_1P_3P_5P_6,\ P_1P_4P_5P_6,\ P_1P_3P_6^2,\ P_2P_3P_5^2,\ P_2P_4P_5^2,\ P_2P_3P_5P_6\}$

And the set of border forbidden states is:

$\mathcal{M}_B=\{P_1P_4P_6^2,\ P_2P_3P_6^2,\ P_2P_4P_5P_6\}$

So according to algorithm 1, we consider an inequality as follows:

$$k_1m_1+k_2m_2+\ldots+k_6m_6 \le x \qquad (5)$$

Putting the authorized states in the inequality (5) leads to the set of inequalities as follows:

$S_1=\{k_1+k_3+2k_5\le x,\ k_1+k_3+k_5+k_6\le x,\ k_1+k_3+2k_6\le x,\ k_1+k_4+2k_5\le x,\ k_1+k_4+k_5+k_6\le x,\ k_1+k_4+2k_6\le x,\ k_2+k_3+2k_5\le x,\ k_2+k_3+k_5+k_6\le x,\ k_2+k_3+2k_6\le x,\ k_2+k_4+2k_5\le x,\ k_2+k_4+2k_6\le x\}$

And putting the forbidden states in the inequality (5) and converting the smaller equal sign to greater sign leads to the set of inequalities as follows:

$S_2=\{k_1+k_4+2k_6>x,\ k_2+k_3+2k_6>x,\ k_2+k_4+k_5+k_6>x\}$

Now we must solve the integer linear programming problem composed of the inequalities in S_1 and S_2 and $x>0$ and $k_i\ge 0$ for $i=1,\ldots,6$ where the objective function is minimum $(k_1+k_2+\ldots+k_6+x)$. The answer for this example is as follow:

$k_1=0,\ k_2=1,\ k_3=1,\ k_4=2,\ k_5=0,\ k_6=1,\ x=3$

So the inequality (5) is obtained as follows:

$m_2+m_3+2m_4+m_6\le 3$

Enforcing this inequality on the system prevent it from entering the forbidden states. So we enforce this constraint on the system using the idea in [9]. The incidence matrix of this system is as follows:

$$W_P=\begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix}$$

And $L=[0\ 1\ 1\ 2\ 0\ 1]$

So, $W_c=[-1\ -1\ 0\ 0\ 1]$

Therefore the incidence matrix of the controlled model is as follows:

$$W=\begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \\ -1 & -1 & 0 & 0 & 1 \end{bmatrix}$$

The controlled model of this system is shown in figure (3).

In this figure, the control place and the related arcs are shown by gray color.

As it is obvious in figure 3, we add a control place to the system for preventing 3 forbidden states. But by the previous methods we must add three control places to the system which is a reason for capability of this method.

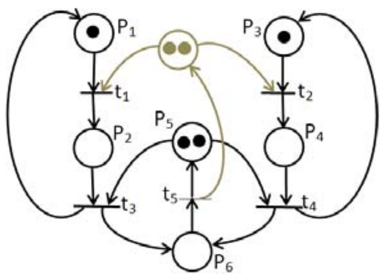

Fig 3. The controlled system in example 1

IV. Conclusion

In this paper we have dealt with the problem of adding extra control places in non safe PN and introduced an algorithm for reducing the number of control places. For this reduction, we have considered an integer linear programming problem which consist a set of inequalities that some of them verify the authorized states and the others violate the forbidden states. Solving this problem may give us a constraint that enforcing it on the system, prevents the system from entering a group of forbidden states. But by the previous methods, we must add a large number of control places to the system.

References

[1] P. Ramadge, W. Wonham, "Modular feedback logic for discrete event systems," *SIAM Journal of Control and Optimization*, 1987, 25(5), 1202–1218.

[2] P. Ramadge, W. Wonham, "The control of discrete event systems," *Dynamics of discrete event systems [Special issue]. Proceedings of the IEEE*, 1989, *77*(1), 81–98.

[3] A. Giua, Petri Nets as Discrete Event Models for Supervisory Control. Ph.D. Thesis, 1992.

[4] J. Moody, P. Antsaklis, "Petri net supervisor for DES with uncontrollable and unobservable transition," *IEEE Trans. Automatic Control*, 2000, 45(3): 462-476.

[5] B. Krogh, L. Holloway, "Synthesis of feedback control logic for discrete manufacturing systems," *Automatica*, 1991, vol. 27, no. 4, pp. 641–651.

[6] L. Holloway, X. Guan, and L. Zhang, "A Generalisation of state avoidance Policies for Controlled Petri Nets," *IEEE Trans. Autom. Control*, 1996, AC-41, 6, 804- 816.

[7] Dideban, A., & Alla, H. (2006). Solving the problem of forbidden states by feedback control logical synthesis. The 32nd Annual Conference of the IEEE Industrial Electronics Society, Paris, FRANCE.

[8] A. Ghaffari, N. Rezg, and X. Xie, "Design of live and maximally permissive petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, 2003, *19*(1).

[9] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of petri nets based on place invariants," *Automatica*, 1996, 32(1), 15–28.

[10] A. Giua, F. DiCesare, M. Silva, Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions. *In Proc. IEEE int. conf. on systems, man, and cybernetics*, 1992, pp. 974–799.

[11] A. Dideban, and H. Alla, "From forbidden state to linear constraints for the optimal supervisory control," *Control Engineering and applied Informaics* (*CEAI*), 2005, 7(3), 48-55.

[12] A. Dideban, and H. Alla, "Reduction of Constraints for Controller Synthesis based on Safe Petri Nets," *Automatica,* 2008, 44(7): 1697-1706.

[13] A. Dideban, M. Zareiee, and H. Alla, Controller synthesis with very simplified linear constraints in PN model. *The 2nd IFAC workshop on Depebdable Control of Discrete Systems*, June 10-12, Bari, Italy, 2009, 265-270.

[14] R. Kumar, L. Holloway, "Supervisory control of deterministic Petri nets with regular specification languages," *IEEE Trans. Automatic Control*, 1996 41(2):245-249.