Reduction of Constraints for Controller Synthesis based on Safe Petri Nets

Abbas Dideban, Hassane Alla

Reduction of Constraints for Controller Synthesis based on Safe Petri Nets

Abbas DIDEBAN[*a], Hassane ALLA[b]

Abstract

In this paper, we present an efficient method based on safe Petri nets to construct a controller. A set of linear constraints allows forbidding the reachability of specific states. The number of these so-called forbidden states and consequently the number of constraints are large and lead to a large number of control places. A systematic method to reduce the size and the number of constraints for safe Petri Nets is offered. By using a method based on Petri nets invariants, maximal permissive controllers are determined.

Key words: Discrete Event Systems (DES), Petri Nets, Supervisory control, Controller synthesis, Forbidden states

[*] Corresponding author. Phone +982313320036, Fax +982313320036
[a] Electrical Eng. Department, Semnan University, Semnan, IRAN, (e-mail: adideban@semnan.ac.ir)
[b] Gipsa Lab, ENSIEG, BP46, 38402 Saint-Martin d'Hères, FRANCE (e-mail: hassane.alla@inpg.fr)

1. Introduction

Supervisory control theory is essentially a theory for restricting the behavior of the plant to satisfy a "safety specification" that specifies which evolutions of the plant should not be allowed. The theory of Ramadge and Wonham (1987; 1989) is based on the modeling of the systems using formal languages and finite automata. However, the great number of states representing the behavior of system, and the lack of structure in the model, limit the possibility of developing an effective algorithm for the analysis and the synthesis of real systems. To solve these problems, several methods of controller synthesis based on Petri Nets (PNs) were proposed. PNs are a suitable tool to study Discrete Event Systems (DES) due to its capability in modeling and its mathematical properties. Very active research in the field of the controller synthesis for DES was born during the last decade (Roussel and Giua 2005; Giua et Xie 2005; Basile et al. 2006).

In (Yamalidou and Moody 1996), (Moody and Antsaklis 2000) and (Basile et al. 2006), the authors use the marking invariants to determine algebraically the incidence matrix of the supervisor PNs model. This method is very simple to be used. However, if some transitions are uncontrollable, it does not give the maximal permissive solution. In the method presented in (Basile et al. 2006) the authors used the structural controllability condition which is only a sufficient condition for having a controllable model. This technique presents two other disadvantages: 1) it is not always possible to describe the specifications by constraints and, 2) the number of constraints can be very large.

The control synthesis consists in preventing from forbidden states. These states may be deduced from specifications and can also be deadlock states. A method to minimize the addition of PN places is proposed in (ZhiWu and Zhou 2004), it is based on elementary siphons. There are some drawbacks in their study. Firstly, one can see that it is based on the computation of minimal siphons and secondly the proposed method is not generally optimal. A third problem is that uncontrollable transitions cannot be considered. In (Uzam 2002; Ghaffari 2003b), the authors proposed a method for solving the problems of forbidden states by the theory of regions. The advantage of this method is its generality for non-safe PNs. However, there are some drawbacks for this method, too:

-Generally, the number of control places is close to the number of border forbidden states.

- The computation time for solving the set of integer equations can be very large.

In (Giua et al. 1992), it is shown that it is possible to use linear constraints to specify forbidden states for safe and conservative PNs. The proposed approach is based on the equivalence between the set of forbidden states and the set

of linear constraints deduced from it. Using the invariants technique presented in (Yamalidou and Moody 1996), allows building a set of control places, which constitutes the optimal controller. However, the number of forbidden states, and consequently, the number of constraints, are large and leads to a large number of control places. In (Giua et al. 1992), it is also shown that some constraints can be replaced by a single one; however, there is no systematic method to calculate the simplified constraints in a general case. The method comes from the linear constraints, which can be simplified taking the PNs structural properties into account.

In (Dideban and Alla 2005), a systematic method has been presented to reduce the number of constraints for safe and conservative PNs. The equations deduced from P-invariants property in conservative PNs are used for simplification. This method needs to construct the set of possible states which is more expensive than the set of reachable states.

In this paper, we relax the property of conservative PNs. Then, a method is proposed to reduce the number of linear constraints for safe PNs. The advantage of this method is that the time and memory space for simplification are less than those presented in (Dideban and Alla 2005). In our approach, we use constraints which are equivalent to forbidden states. These constraints can be calculated in two different ways. They can be given directly as specifications or they can be deduced thanks to the Kumar approach (Kumar and Holloway, 1996).

In this paper, the important concept of *over–state* will be defined. This concept corresponds to a set of markings which has the same property. This idea will help us to build the simplest constraints, which forbid a greater number of states. A property for the existence of the maximal permissive controller will be analytically proved. In some very particular cases of non conservative PNs, the optimal solution does not exist. We show that this approach allows highlighting this problem in a simple way. This important concept can be used in other approaches.

In our approach, as in (Dideban and Alla 2005), we use the Reachability Graph (RG) as an intermediate step for calculating the controller. Although the complexity of the computation of RG is exponential, this calculation is performed off-line. Moreover, the implemented final controller is a PN model, whose size is very close to the initial model. Generally, few control places are added.

The rest of this paper is organized as follows: In Section 2, the motivation and the fundamental definitions will be presented and illustrated via an example. In Section 3, the idea of passage from forbidden states to the linear constraints will be introduced. The concept of over-state and the basic idea of the simplification will be presented in Section 4. The calculation of the maximal permissive controller will be described in Sections 5. Finally, the conclusion is given in the last section.

2. Preliminary presentation

In this paper, it is supposed that the reader is familiar with the PNs basis (David and Alla 2005) and the theory of supervisory control (Ramadge and Wonham 1987; 1989). In this section, we present only the notations and definitions which will be used later.

A PN is represented by a quadruplet $R = \{P, T, W, M_0\}$ where P is the set of places, T is the set of transitions, W is the incidence matrix and M_0 is the initial marking. This PN is assumed to be safe; the marking of each place is Boolean.

Definition 1: The set $\{0,1\}^N$ represents all the Boolean vectors of dimension N.

❑

A marking of a safe PN containing N places is a vector of the set $\{0,1\}^N$.

The set of the marked places of a marking M is given by a function support defined as below:

Definition 2: The function Support(X) of a vector $X \in \{0,1\}^N$ is:
Support(X) = the set of marked places in X.

❑

The support of vector $M_0{}^T = [1, 0, 1, 0, 0, 1, 0]$ is:

Support $(M_0) = \{P_1P_3P_6\}$ or more simply:

Support $(M_0) = P_1P_3P_6$

To simplify the notation of the formal expressions, we will use the support of a marking instead of its corresponding vector.

$\mathcal{M}_R$ denotes the set of PN reachable markings. In $\mathcal{M}_R$, two subsets could be distinguished: the set of authorized states $\mathcal{M}_A$ and the set of forbidden states $\mathcal{M}_F$. The set of forbidden states correspond to two groups: 1) the set of reachable states ($\mathcal{M}_{F'}$) which either do not respect the specifications or are deadlock states. 2) the set of states for which the occurrence of uncontrollable events leads to states in $\mathcal{M}_{F'}$.

The set of authorized states are the reachable states without the set of forbidden states:

$$\mathcal{M}_A = \mathcal{M}_R \setminus \mathcal{M}_F$$

Among the forbidden states, an important subset is constituted by the border forbidden state denoted as $\mathcal{M}_B$.

Definition 3: Let $\mathcal{M}_B$ be the set of border forbidden state:

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

Where Σ_c is the set of controllable transitions

❑

We will use the following example in order to illustrate the definitions and the results developed in this paper.

Consider a system composed of two machines Ma_1 and Ma_2 which can work independently. The starting and the end of the tasks on these machines are respectively realized by controllable events c_1 and c_2, and by uncontrollable events f_1 and f_2. When machine Ma_1 ends its task on a part, it stays available for a new task while machine Ma_2 has to

transfer its produced part in a buffer before beginning a new task (event b_2). Both machines are activated simultaneously (event *start*) but each of them can be inactivated separately (events sp_1 and sp_2). The specifications impose a sequence of the events f_1 and b_2. An elementary production is a result of a process on a part by Ma_1 followed by another process by Ma_2. This production is repeated in a cyclic way. The system can be started by a start command and can be stopped by a stop command. At the end, the production process on a part must be completed. For restart, we need to initialization of the controller. The process and specifications models are represented in Figure 1. They are non conservative PNs.

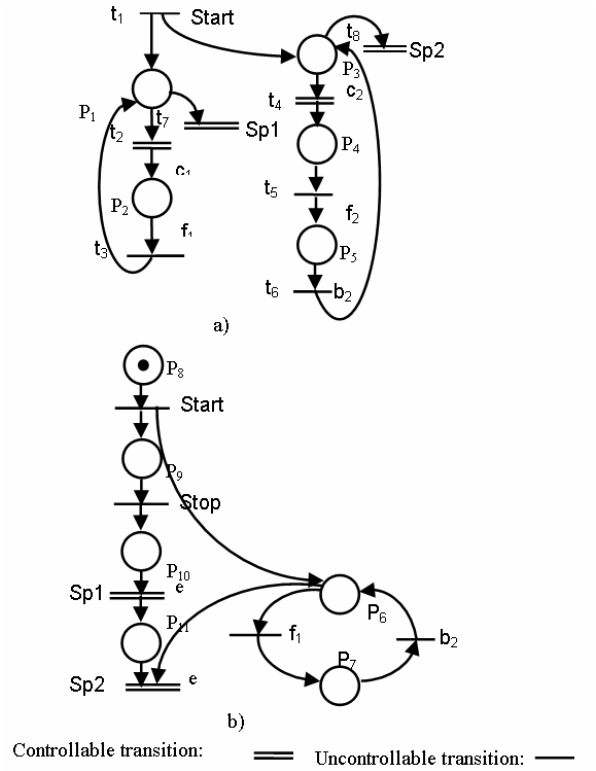


Fig. 1. PN model of the a) Process b) specification

The synchronous composition between the models of process and the model of specifications is given by a safe PN in Figure 2.

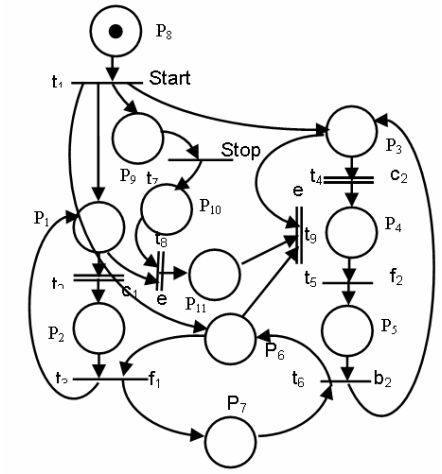


Fig. 2. PN model of the system coupled with its specification

The existence of uncontrollable events leads to the existence of forbidden states. For example when the system is in state M_5, it is possible to fire the uncontrollable event f_1, while it is not authorized by the specifications. This state is a forbidden state. The set of forbidden states can be determined by the algorithm established by Kumar and Holloway (1996).

Figure 3 gives the reachability Graph of the PN presented in Figure 2. The forbidden states are indicated in dark gray and the authorized states in white. The construction of the reachability graph is stopped when a forbidden state is reached.

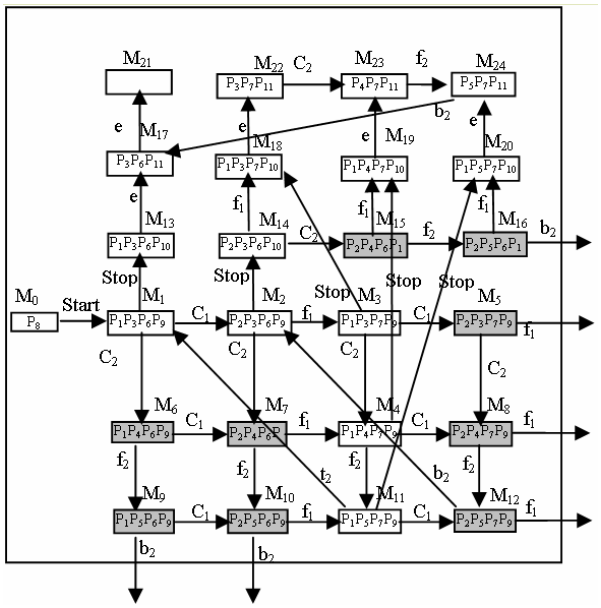


Fig. 3 Reachability graph

From the set of forbidden states $\mathcal{M}_F = \{M_5, M_6, M_7, M_8, M_9, M_{10}, M_{12}, M_{15}, M_{16},\ldots\}$, we can construct the set of border forbidden states $\mathcal{M}_B$

$\mathcal{M}_B = \{M_5, M_6, M_7, M_8, M_{12}, M_{15}\}$

In a conservative and safe PN, the inequality $m_1 + m_4 + m_6 + m_9 \leq 3$ forbids only the state $P_1P_4P_6P_9$. (Giua et al. 1992). In this situation, for N forbidden states, we will need N linear constraints. The complexity of the controller model increases extremely when the number of forbidden states increases for we need one control place for each constraint (Yamalidou and Moody 1996). In this paper, we propose a method to reduce the number and the size of the linear constraints for a given set of forbidden states. We give the necessary and sufficient condition for having a maximal permissive controller in the case of non conservatives PNs. To achieve this goal, we need to introduce the important concept of "*over-state*". In this paper we use a hypothesis that is presented below:

Hypothesis 1: All of the events are independent.

❑

3. From forbidden states to linear constraints

Let M_i ($M_i^T = [m_1, m_2, \ldots, m_N]$) be a forbidden state[1] in set $\mathcal{M}_B$ and Support(M_i) = $\{P_{i1}\ P_{i2}\ P_{i3}\ldots\ P_{in}\}$ the set of marked places of M_i. From a forbidden state, a linear constraint can be constructed (Giua et al. 1992).

The linear constraint deduced from the forbidden state M_i is given below. The state M_i does not verify this relation. Therefore, by applying this relation, M_i will be forbidden.

$$\sum_{k=1}^{n} m_{ik} \leq n - 1$$

Where n = Card [Support (M_i)] is the number of marked places of M_i, and m_{ik} is the marking of place P_{ik} of state M_i.

Let M ($M^T = [m_1, m_2, \ldots, m_N]$) be a general marking and M_i be a forbidden state. The constraint (forbidding state M_i) is denoted by c_i and can be rewritten in the following form:

$$M_i^T . M \leq \text{Card [Support } (M_i)] - 1$$

For example if:

$$M_i^T = [0, 1, 1, 0, 0, 0, 1] \Rightarrow \text{Card [Support } (M_i)] = 3$$

$$\Rightarrow m_2 + m_3 + m_7 \leq 2 \qquad (1)$$

Verifying Relation 1 is equivalent to forbid state M_i when the PN model is conservative. However, in a safe PN not necessarily conservative, this equivalence is not always true. This problem will be discussed later. This equivalence is necessary to obtain the optimal supervisor.

4. Simplification by using over-state concept

4.1. Definition of an over - state

The concept of over-state is very important in this paper. An over-state can represent a complete state or a part of this one. In the example of the two machines, $P_2P_3P_6P_9$ is a complete state that represents the situation of both machines and the specifications. P_2P_3 is an over-state of this state that represents a partial state of the system. We have noted that a state can be forbidden by a linear constraint. In the same way, it is possible to forbid an over-state by its corresponding constraint.

Definition 4: Let $M_2 = P_{21}\ P_{22}\ \ldots P_{2m}$ be an accessible state, $M_1 = P_{11}\ P_{12}\ \ldots P_{1n}$ will be an over-state of M_2 if:

$$M_1 \leq M_2$$

❑

For example $M_1 = P_1P_3$ is an over-state of $M_2 = P_1P_3P_6P_9$.

The name "*over-state*" is used because the constraint corresponding to an over-state holds the state's constraint. For example, the constraint $m_4 + m_6 \leq 1$ that corresponds to the over-state $M_1 = P_4P_6$ holds both following constraints:

$$m_1 + m_4 + m_6 + m_9 \leq 3$$
$$m_2 + m_4 + m_6 + m_9 \leq 3$$

These two constraints forbid states $M_6 = P_1P_4P_6P_9$ and $M_7 = P_2P_4P_6P_9$. P_4P_6 is an over-state of both states $P_1P_4P_6P_9$ and $P_2P_4P_6P_9$ which could be verified by $M_1 \leq M_6$ and $M_1 \leq M_7$. Thus by using only the constraint $m_4 + m_6 \leq 1$, both states M_6 and M_7 will be forbidden. However, this reduction is not always simple; it is possible that the simplified constraint forbids also some authorized states. We present below a method of simplification which guarantees that the constraints forbid only the forbidden states.

Remark 2: With each over-state b_i, we associate a constraint c_i in the following way:

$$b_i = (P_{i1}P_{i2}P_{i3}\ldots P_{in}) \Rightarrow c_i = (P_{i1}P_{i2}P_{i3}\ldots P_{in}, n\text{-}1)$$

That means:

$$m_{i1} + m_{i2} + \ldots + m_{in} \leq n\text{-}1$$

❑

Remark 3: It is possible to use an over-state without taking into account the fact that an authorized state can be forbidden. In that case, the controller would not be maximal permissive.

❑

Remark 4: There are two relations of inclusion, which operate in opposite directions: a set inclusion and a marking inclusion. Let $M_1 \leq M_2$:

1) The set of the marked places in the over-state M_1 is included in the set of the marked places in the state M_2.
2) The set of the markings covered by M_1 contains those covered by marking M_2.

❑

***Property* 1:** Let M_1 and M_2 be two vectors of $\{0, 1\}^N$, and c_1 and c_2 be two corresponding constraints. If $M_1 \leq M_2$ (M_1 is an over-state of M_2) and c_1 is true, then c_2 is also true:

$$M_1 \leq M_2 \text{ and } c_1 : M_1^T . M \leq \text{Card[Support}(M_1)] - 1$$
$$\Rightarrow c_2: \quad M_2^T . M \leq \text{Card[Support}(M_2)] - 1$$

Proof:

The PN model is safe then:

$$(M_2^T - M_1^T).M \leq \text{Card[Support}(M_2)] - \text{Card[Support}(M_1)]$$

And:

$$M_2^T . M = (M_2^T - M_1^T + M_1^T).M = (M_2^T - M_1^T).M + M_1^T.M$$

By using the constraint c_1, we have:

$$(M_2^T - M_1^T).M + M_1^T.M \leq (\text{Card[Support}(M_2)] - \text{Card[Support}(M_1)]) + \text{Card[Support}(M_1)] - 1$$
$$\Rightarrow \quad M_2^T . M \leq \text{Card[Support}(M_2)] - 1$$

❑

4.2. Set of over-states

We have noted that to forbid a state, it is enough to forbid its over-state, but which over-state? This question will

[1] When there is no ambiguity, the word *border* will be omitted.

be answered in the sequel. To achieve this goal, we need to construct the set of over-states for the forbidden states.

Firstly, we calculate the set of over-states for each state and then the union of all over-states gives the final set.

Definition 5: Let $M_i = \{P_{i1}P_{i2}P_{i3} \ldots P_{in}\}$ be a state of the system. The set of the over-states of M_i, denoted by M_i^{over}, is equal to the set of the subsets of M_i without the empty set.

❑

For example, the state $M_1 = P_1P_4P_6P_9$ give:

$$M_1^{over} = \{P_1, P_4, P_6, P_9, P_1P_4, P_1P_6, P_1P_9, P_4P_6, P_4P_9, P_1P_4P_6, P_1P_4P_9, P_1P_6P_9, P_4P_6P_9, P_1P_4P_6P_9\}$$

Among, the set of forbidden states in $\mathcal{M}_F$, only the border states have to be considered in the controller synthesis. Let $\mathcal{M}_B$ be this set and B_1 be the set of over-states of $\mathcal{M}_B$.

$$B_1 = \bigcup_{i=1}^{Card(M_B)} M_i^{over}$$

4.3. Basic idea to build the minimal set of constraints

For a given set and a property, we can define three disjoint sub-sets:

1) The set where each element verifies this property

2) The set where each element does not verify this property, and

3) The set which is indifferent to this property.

The third set is important and will be used advantageously to improve the simplifications.

Definition 6: Let E_1 and E_2 be two sets included in a set G and hold:

$$E_1 \bigcap E_2 = \phi$$
$$E_1 \bigcup E_2 = E$$
$$E \bigcup \bar{E} = G$$

$\bar{E}$ is the complementary set of E in G.

V is an element of G.

A property P according to V is true if $V \in E_1$ and false if $V \in E_2$.

❑

This property is not defined if $V \in \bar{E}$, it can then be said that this property is true if $V \notin E_2$.

We will use this definition on the set of states to achieve our goal. Let E_1 be the set of the forbidden states and E_2 the set of the authorized states and let P be the forbidding property. The state V can be forbidden if it is not in E_2. This means that the states which are not accessible could be forbidden. This consideration will make the constraints to be further simplified. This idea is similar to the concept of don't care states that are used in the minimization of combinatorial and sequential logic. In logic circuits don't care states are the states that are not reachable because of the input variables or initial states. In PN models, non reachable states are the states that are not accessible from the initial state.

In Property 1, it was shown that one over-state can cover a great number of states. Therefore, we can forbid an over-state if it does not cover any authorized state.

Our objective is to find a method to reduce the number and the bound of the constraints. For that, we build the set of all over-states of the border forbidden states. This set will be calculated by removing all authorized over-states from it. The minimal set of constraints will then be obtained. Finally, the best choice will be established.

The different steps formalizing this approach are presented in the following section.

4.4. Building the reduced set of over-states

It is possible to build two sets of over-states; a set of the authorized over-states A_1, and that of the forbidden states B_1. It is obvious that no over-state of A_1 must be forbidden. Thus it is necessary to remove from set B_1, all over-states which are in A_1. This gives set B_2:

$$B_2 = B_1 \setminus A_1$$

Remark 5: From the implementation point of view, it is not necessary to construct A_1. The set $\mathcal{M}_A$ is directly used.

Property 2: Let B_1 be the set of over-states of $\mathcal{M}_B$ and A_1 be the set of over-states of $\mathcal{M}_A$ and:

$$B_2 = B_1 \setminus A_1$$

The markings verifying the set of constraints C_2 (equivalent to B_2) correspond to the complete set of authorized states.

❑

The proof of this property is obvious.

In set B_2, it often possible to find couple of states M_1 and M_2 such that $M_1 \leq M_2$ (M_1 is an over-state of M_2). In that case, M_2 must be removed. It is a redundant state, and set B_3 is then defined formally as follows:

$$B_3 = B_2 - \{M_{2i} \in B_2 \mid \exists\, M_{2j} \in B_2, M_{2i} \geq M_{2j}\}$$

B_3 is the minimal set of over-states to be forbidden.

For the example, from Figure 2, the sets of border forbidden states and authorized states are:

$\mathcal{M}_B = \{P_1P_4P_6P_9, P_2P_4P_6P_9, P_2P_3P_7P_9, P_2P_4P_7P_9, P_2P_5P_7P_9, P_2P_4P_6P_{10}\}$

$\mathcal{M}_A = \{P_6P_8, P_3P_7P_{11}, P_4P_7P_{11}, P_5P_7P_{11}, P_3P_6P_{11}, P_1P_3P_6P_9, P_2P_3P_6P_9, P_1P_3P_7P_9, P_1P_4P_7P_9, P_1P_5P_7P_9, P_1P_3P_6P_{10}, P_2P_3P_6P_{10}, P_1P_3P_7P_{10}\}$

Sets A_1, B_1, B_2 and B_3 are then calculated as follows:

$B_1 = M_1^{over} \cup M_2^{over} \cup M_3^{over} \cup M_4^{over} \cup M_5^{over} \cup M_6^{over} =$
$\{ P_1 , P_2, P_3, P_4 , P_5 , P_6 , P_7 , P_9, P_{10}, P_1P_4 , P_1P_6 , P_1P_9, P_4P_6 , P_4P_9, P_6P_9, P_2P_4 , P_2P_6 , P_2P_9, P_2P_3 , P_2P_7 , P_3P_7, P_3P_9 , P_7P_9, P_4P_7 , P_2P_5 , P_5P_7 , P_5P_9, P_2P_{10} , P_4P_{10} , P_6P_{10}, P_1P_4P_6, P_1P_4P_9, P_1P_6P_9, P_4P_6P_9, P_2P_4P_6, P_2P_4P_9, P_2P_6P_9, P_2P_3P_7, P_2P_3P_9, P_2P_7P_9, P_3P_7P_9, P_2P_4P_7, P_2P_7P_9, P_4P_7P_9, P_2P_5P_7, P_2P_5P_9, P_5P_7P_9, P_2P_4P_{10}, P_2P_6P_{10}, P_4P_6P_{10},$

$P_1P_4P_6P_9$, $P_2P_4P_6P_9$, $P_2P_3P_7P_9$, $P_2P_4P_7P_9$, $P_2P_5P_7P_9$, $P_2P_4P_6P_{10}$}

A_1 ={ P_1 , P_2 , P_3 , P_4 , P_5 , P_6, P_7 , P_8, P_9, P_{10}, P_{11}, P_6P_8, P_3P_7, P_3P_{11} , P_7P_{11} , P_4P_7, P_4P_{11} , P_5P_7 , P_5P_{11} ,..., $P_3P_7P_{11}$, $P_4P_7P_{11}$, $P_5P_7P_{11}$, $P_3P_6P_{11}$, $P_1P_3P_6P_9$, $P_2P_3P_6P_9$, $P_1P_3P_7P_9$, $P_1P_4P_7P_9$, $P_1P_5P_7P_9$, $P_1P_3P_6P_{10}$, $P_2P_3P_6P_{10}$, $P_1P_3P_7P_{10}$}

B_2= { P_1 , P_2, P_3, P_4 , P_5 , P_6 , P_7 , P_9, P_{10}, P_1P_4 , P_1P_6 , P_1P_9, P_4P_6 , P_4P_9, P_6P_9, P_2P_4 , P_2P_6 , P_2P_9, P_2P_3 , P_2P_7 , P_3P_7, P_3P_9 , P_7P_9, P_4P_7 , P_2P_5 , P_5P_7 , P_5P_9, P_2P_{10} , P_4P_{10} , P_6P_{10}, $P_1P_4P_6$, $P_1P_4P_9$, $P_1P_6P_9$, $P_4P_6P_9$, $P_2P_4P_6$, $P_2P_4P_9$, $P_2P_6P_9$, $P_2P_3P_7$, $P_2P_3P_9$, $P_2P_7P_9$, $P_3P_7P_9$, $P_2P_4P_7$, $P_2P_7P_9$, $P_4P_7P_9$, $P_2P_5P_7$, $P_2P_5P_9$, $P_5P_7P_9$, $P_2P_4P_{10}$, $P_2P_6P_{10}$, $P_4P_6P_{10}$, $P_1P_4P_6P_9$, $P_2P_4P_6P_9$, $P_2P_3P_7P_9$, $P_2P_4P_7P_9$, $P_2P_5P_7P_9$, $P_2P_4P_6P_{10}$}

B_3={P_4P_6 , P_2P_4 , P_2P_7 , P_2P_5 , P_4P_{10} , $P_1P_4P_6$, $P_4P_6P_9$, $P_2P_4P_6$, $P_2P_4P_9$, $P_2P_3P_7$, $P_2P_3P_9$, $P_2P_7P_9$, $P_3P_7P_9$, $P_2P_4P_7$, $P_2P_7P_9$, $P_2P_5P_7$, $P_2P_5P_9$, $P_2P_4P_{10}$, $P_4P_6P_{10}$, $P_1P_4P_6P_9$, $P_2P_4P_6P_9$, $P_2P_3P_7P_9$, $P_2P_4P_7P_9$, $P_2P_5P_7P_9$, $P_2P_4P_6P_{10}$} = {P_4P_6, P_2P_4, P_2P_7, P_2P_5, P_4P_{10}}

Remark 6: In reality we don't need to construct A_1. It is possible calculate B_2 from B_1 and $\mathcal{M}_A$.

❑

5. Controller synthesis

5.1. Maximal permissive controller

In the previous section, we have determined the set B_3, which is the set of over-states that must be forbidden. In the two following sections, we present the necessary and sufficient conditions to design a maximal permissive controller.

With each over-state of B_3, we associated a constraint in the following way:

$b_i = (P_{i1}P_{i2}P_{i3}\ldots P_{in}) \Leftrightarrow c_i = (P_{i1}P_{i2}P_{i3}\ldots P_{in}, n\text{-}1)$

Let C_3 be the set of these constraints for the example:

C_3 = {(P_4P_6,1), (P_2P_4, 1), (P_2P_7, 1), (P_2P_5, 1), (P_4P_{10}, 1)}

This set C_3 defines the set of non-forbidden states, denoted as $\mathcal{M}_E$. Now the objective is to compare the set of authorized states $\mathcal{M}_A$ and $\mathcal{M}_E$.

Remark 7: Constraint c_i and over-state b_i are equivalent as shown above.

❑

Definition 7: Let $B_3 = \{b_1, b_2, \ldots, b_m\}$ be the set of simplified over-states and $\mathcal{M}_B = \{M_1, M_1, \ldots, M_N\}$ be the set of border forbidden states. The relation R: $\mathcal{M}_B \times B_3 \rightarrow \{0, 1\}$ is as:

$$R(M_i, b_j) = \begin{cases} 1 & b_j \le M_i \ (b_j \text{ is over-state of } M_i) \\ 0 & \text{if not} \end{cases}$$

The covering of a marking is an integer number:

$$Cv(M_i) = \sum_{j=1}^{m} R(M_i, b_j)$$

❑

$Cv(M_i) \ge 1$ means that forbidden state M_i is covered by at least one over-state.

Property 3: The set of non forbidden state $\mathcal{M}_E$ is equal to the set of authorized state $\mathcal{M}_A$ if and only if:

$$\forall M_i \in \mathcal{M}_B \quad Cv(M_i) \ge 1$$

❑

Proof:

Necessary Condition:

Assume that $\mathcal{M}_A = \mathcal{M}_E$, we prove that:

$$\forall M_i \in \mathcal{M}_B \ Cv(M_i) \ge 1$$

If $\exists M_i \in \mathcal{M}_B / Cv(M_i) = 0 \Rightarrow R(M_i, b_j) = 0 \ \forall b_j \in B_3$,

There is not any constraint c_j deduced from b_j that forbids M_i. Then

$$M_i \in \mathcal{M}_E$$

However, M_i is a forbidden state and, $M_i \notin \mathcal{M}_A$,

Then $\mathcal{M}_A \neq \mathcal{M}_E$ that it is not true.

Sufficient condition:

Assume that $\forall M_i \in \mathcal{M}_B \ Cv(M_i) \ge 1$, we prove:

$\mathcal{M}_A = \mathcal{M}_E$

$\forall M_i \in \mathcal{M}_B$, $Cv(M_i) \ge 1 \Rightarrow$

$\forall M_i \in \mathcal{M}_B \ \exists b_j \in B_3 / R(M_i, b_j) = 1$, ($M_i$ would be forbidden by this constraint)

$\Rightarrow \forall M_i \in \mathcal{M}_B$, $M_i \notin \mathcal{M}_E$ Then :

$$\mathcal{M}_E \subseteq \mathcal{M}_A$$

In addition, according to the method used for the construction of B_3, $\mathcal{M}_A \subseteq \mathcal{M}_E$ (any authorized state is not forbidden)

Then $\mathcal{M}_A = \mathcal{M}_E$

❑

Now, let us illustrate the results established above in the example of Figure 2. Property 3 should initially be checked. For this, we construct a table (Table 1) where the first row represents the set of forbidden states $\mathcal{M}_B$ and the first column is the set of simplified over -states B_3. In the case of our example, these sets are:

$\mathcal{M}_B$ = {$P_1P_4P_6P_9$, $P_2P_4P_6P_9$, $P_2P_3P_7P_9$, $P_2P_4P_7P_9$, $P_2P_5P_7P_9$, $P_2P_4P_6P_{10}$}

B_3 = {P_4P_6, P_2P_4, P_2P_7, P_2P_5, P_4P_{10}}

$c_j \downarrow$ $M_i \rightarrow$	$P_1P_4P_6P_9$	$P_2P_4P_6P_9$	$P_2P_3P_7P_9$	$P_2P_4P_7P_9$	$P_2P_5P_7P_9$	$P_2P_4P_6P_{10}$
P_2P_4	0	1	0	1	0	1
P_2P_5	0	0	0	0	1	0
P_4P_6	1	1	0	0	0	1
P_2P_7	0	0	1	1	1	0
P_4P_{10}	0	0	0	0	0	1
$Cv(M_i)$	1	2	1	2	2	3

Tab. 1. Function $R(c_j, M_i)$ and $Cv(M_i)$

This table shows that $\forall M_i \in \mathcal{M}_B \ Cv(M_i) \ge 1$, and thus the set of non forbidden states $\mathcal{M}_E$ is equal to the set of authorized states $\mathcal{M}_A$.

We will see that this is not always the case. For that, we take the example presented in (Bratosin et al. 2005) . It is a system made up of two machines M_1 and M_2. The beginnings of the tasks are denoted by the controllable events c_1 and c_2 and the ends are synchronized by the uncontrollable event f. The specification authorizes the occurrence of the event f only once. The PN model $\mathcal{R}_e$ of the closed-loop operation for this system is presented in Figure 4.

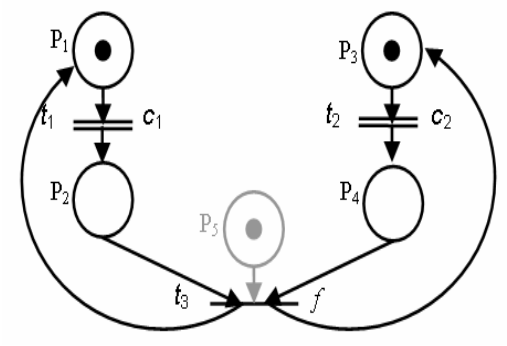


Fig. 4. Closed loop PN Model in case of non optimal supervisor

The sets of the authorized and forbidden states are presented below:

$\mathcal{M}_B = \{P_1P_4, P_2P_3\}$

$\mathcal{M}_A = \{P_1P_3P_5, P_2P_3P_5, P_2P_4P_5, P_1P_4P_5, P_1P_3\}$

$B_1 = \{P_1, P_2, P_3, P_4, P_1P_4, P_2P_3\}$

$A_1 = \{P_1, P_2, P_3, P_4, P_5, P_1P_3, P_1P_5, P_3P_5, P_2P_3, P_2P_5, P_2P_4, P_4P_5, P_1P_4, P_1P_3, P_1P_3P_5, P_2P_3P_5, P_2P_4P_5, P_1P_4P_5\}$

$B_2 = \{ \overline{P_1}, \overline{P_2}, \overline{P_3}, \overline{P_4}, \overline{P_1P_4}, \overline{P_2P_3} \} = \phi \Rightarrow B_3 = \phi$

$\forall M_i \in \mathcal{M}_B \quad Cv(M_i) = 0$ and $\mathcal{M}_E = \mathcal{M}_{\mathcal{R}_e}$ (All accessible states).

Here is a case where $\mathcal{M}_A \subset \mathcal{M}_E$. Then it is not possible to construct the maximal permissive controller for this model of system.

This type of behavior is rarely met in real cases. We have built it artificially. Moreover, generally as in this example, for the case of non conservative and safe PN, we can modify the PN model by adding one place in order to have a conservative PN. For example, It is sufficient to add place P_6 after transition t_3 as it is shown in Figure 5. Now the set of places P_5 and P_6 belongs to the invariant: $m_5 + m_6 = 1$ and in this case we are able to construct a maximal permissive controller.

5.2. Final covering

After the simplifications presented above, it is possible to choose the simplest constraints covering all forbidden states. In the result of the last step, the same forbidden state can be covered by several over-states. The rules to choose the final over-states are similar to the rules of the Quine-McCluskey method to simplify the logical expressions (Morris Mano 2001). To choose the final results, table 1 is used and modified in table2.

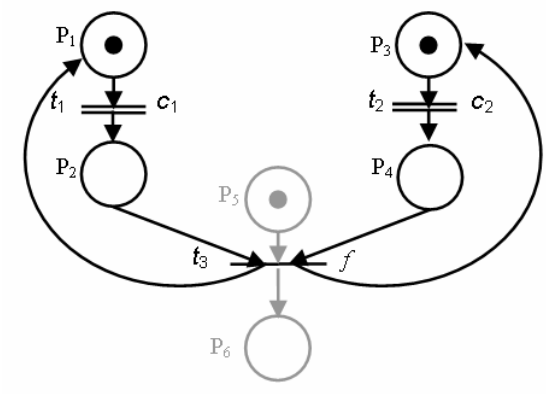


Fig. 5. Modified model of Figure 4

$c_j \downarrow$ $M_i \rightarrow$	$P_1P_4P_6P_9$	$P_2P_4P_6P_9$	$P_2P_3P_7P_9$	$P_2P_4P_7P_9$	$P_2P_5P_7P_9$	$P_2P_4P_6P_{10}$	B_4
P_2P_4	0	1	0	1	0	1	-
P_2P_5	0	0	0	0	1	0	-
P_4P_6	1	1	0	0	0	1	**1**
P_2P_7	0	0	1	1	1	0	**1**
P_4P_{10}	0	0	0	0	0	1	-
$Cf(M_i)$	1	1	1	1	1	1	

Tab. 2. Function $R(c_j, M_i)$ and $Cf(M_i)$

To choose the *minimal set of constraints*, denoted by B_4, firstly it is necessary to choose the over-state for which there exists a forbidden state that can be covered only by this over-state ($Cv(M_i) = 1$). If such over-states are found, we mark all the corresponding forbidden states in line $Cf(M_i)$. This line corresponds to the final covering. Then if a forbidden state is covered by two or several over-states, it is necessary to choose the over-state which covers the most non selected forbidden states. In the case of equality, the simplest over-state will be selected. These ideas are formalized in the algorithm 1 presented in appendix I.

Corollary 1: The set of the non forbidden states $\mathcal{M}_E$ defined by the set of the constraints deduced from B_4 is equal to M_A if and only if:

$\forall M_i \in \mathcal{M}_B \quad Cf(M_i) = 1$

❑

This corollary means that it is necessary for each forbidden state to be covered at least by one over-state. When this is verified, the maximal permissive controller is obtained.

5.3. Control places

The set of the constraints equivalent to B_4 is denoted by C_4. To calculate the control places corresponding to each linear constraint, we will use the method developed in (Yamalidou and Moody 1996). This technique based on the PNs *invariant* is recalled briefly below. Let W_R be the incidence matrix of the system (process and specifications). Each place of the controller will add a line to the matrix. Let W_{RC} be the incidence matrix of the PN model corresponding to the controlled system. It is made up of two matrices, the original matrix of system R, W_R and the incidence matrix of the controller, W_C. From the set of constraints C_4, matrix L and constant vector C_bound can be constructed. It is possible to calculate in an algebraic way the incidence matrix of the controller as it is presented below. M_{Ri} is the initial marking of system R and M_{Ci} is the initial marking of the control places. The very simple way to calculate W_C makes this approach very popular.

$$W_C = - L.W_R$$

$$M_{C\_i} = C\_bond - L.M_{R\_i}$$

Let us take again the example of Figure 2, the set of final constraints (C_4) is:

$$m_4 + m_6 \leq 1 \quad , \quad m_2 + m_7 \leq 1$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W_C = - L.W_R$$

$$\Rightarrow \quad W_C = \begin{bmatrix} 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_{C\_i} = C\_bond - L.M_{R\_i}$$

$$\Rightarrow \qquad M_{c1} = 0; \qquad M_{c2} = 1$$

Yamalidou and Moody (1996) showed that if all events are controllable, the controller is maximal permissive. However, if there are uncontrollable events, the extended method presented in (Moody and Antsaklis 2000) does not generally give the optimal solution. The problem exists when a control place is synchronized with a place of the process by an uncontrollable event as indicated in figure 6.

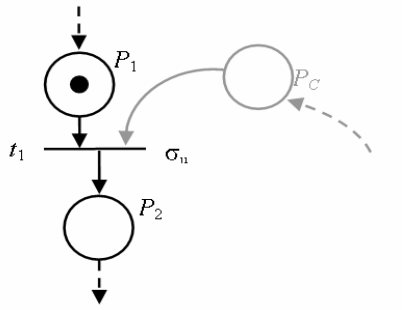


Fig. 6. Control place synchronized with the process by an uncontrollable event

In the case presented in Figure 6, the process cannot always respect the PN firing rules. Suppose that place P_c is not marked and P_1 is marked. Since σ_u is uncontrollable, then transition T_1 is fired even if it is forbidden by the control place. It means that it is possible that the set of reachable states will be greater than the set given by the PN model. According to definition of structural controllable model in (Basile et al. 2006), the model in this case is not controllable. We prove in Property 4 that it is not a necessary condition after applying our method of controller synthesis. After using our method, when the places belong to the process are marked, the control and specifications places will be always marked.

Definition 8: The set of accessible states for controlled system is presented by the set $\mathcal{A}_{RC}$.

❑

We are going to show that if the condition in Corollary 1 is true, the obtained controller is maximal permissive even if uncontrollable transitions exist.

Remark 8: A marking of the set $\mathcal{A}_{RC}$ differs from a marking of $\mathcal{M}_E$ because of the added control places. This is only a coding of these sets. To be able to compare the various sets of states, we will omit the control places for the elements of the set $\mathcal{A}_{RC}$.

❑

Property 4: Let $\mathcal{M}_E$ be the set of authorized states by the constraints deduced from B_4 and let $\mathcal{A}_{RC}$ be the automaton that corresponds to the set of accessible state in the controlled system,

If $\mathcal{M}_E = \mathcal{M}_A$, then $\mathcal{A}_{RC}$ is isomorphic to $\mathcal{M}_E$ and the controller obtained by the invariant approach is maximal permissive.

❑

***Proof*:**

By the invariant approach, we have always:

$$\mathcal{M}_E \subseteq \mathcal{A}_{RC} \qquad (2)$$

Now we show that:

$$\mathcal{A}_{RC} \subseteq \mathcal{M}_E \text{ (knowing that } \mathcal{M}_E = \mathcal{M}_A)$$

Suppose that $\exists\, M_i \in \mathcal{A}_{RC}$ and $M_i \notin \mathcal{M}_E$

$$\Rightarrow \exists\, \sigma_u \in \Sigma_u \text{ and } \exists\, M_j \in \mathcal{M}_E,\ M_j \xrightarrow{\sigma_u} M_i$$

However $\mathcal{M}_A = \mathcal{M}_E \Rightarrow M_j \in \mathcal{M}_A$ and $M_i \notin \mathcal{M}_A$

$M_i \notin \mathcal{M}_A \Rightarrow M_i \in \mathcal{M}_F$ $\qquad (\mathcal{M}_F = \mathcal{M}_R \setminus \mathcal{M}_A)$

It is obvious that: $M_j \xrightarrow{\sigma_u} M_i$ then $M_j \in \mathcal{M}_F$ (definition of forbidden states)

$M_j \in \mathcal{M}_A$ and $M_j \in \mathcal{M}_F$ (contradiction), then

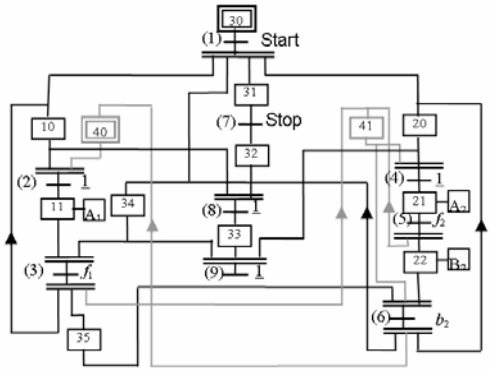
$$\mathcal{A}_{RC} \subseteq \mathcal{M}_E \qquad (3)$$

(2) and (3) $\Rightarrow$ $\mathcal{M}_E = \mathcal{A}_{RC}$ $\Rightarrow$ $\mathcal{A}_{RC} = \mathcal{M}_A$

❑

In the case of our example, the function $Cf(M_i)$ (final covering) is equal to 1 for each $M_i \in \mathcal{M}_B$, therefore $\mathcal{M}_E = \mathcal{M}_A$ (Corollary 1) then the controller is maximal permissive (Property 4). The PN model of the final controller is represented in Figure 7.

It should be noticed that there are some control places with uncontrollable output transitions. However, that never leads to a bad behavior, i.e. when a control place is not



marked; there is at least one non marked input place for this uncontrollable transition, which belongs to the process. Moreover, controllable events c_1 and c_2 have been removed since the control is now performed by the control places. The complete algorithm for controller synthesis is presented in Appendix I. The computation of some sets is of polynomial complexity except for the $\mathcal{M}_B$ over-states computation which is exponential. Fortunately the number of border sates is often small.

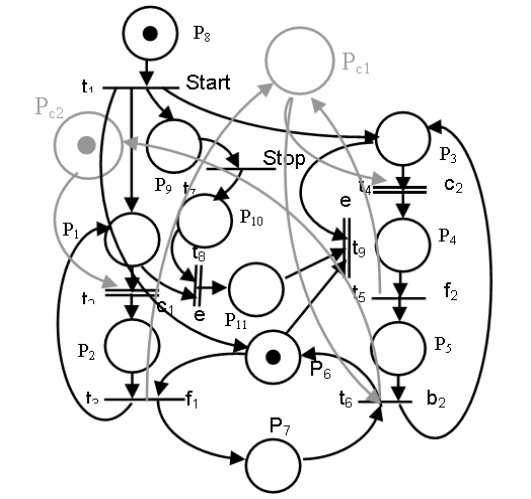


Fig. 7. PN Model in closed loop with control places

5.4. From PNs to SFC models

The controllers have always a deterministic behavior. A given set of inputs corresponds to a unique set of outputs. In this paper we consider an asynchronous functioning, all events are independent and the simultaneous occurring of two independent events is not possible. However in real implementation, due to cycle time in a PLC (Programmable Logic Controller), it is possible to have simultaneous occurring of events. Then, sometimes the controller obtained with our approach can be non deterministic. In that case, the conflicts must be solved for example by making a choice.

In the example of Figure 6, the model is deterministic and there is no conflict. We can transfer directly the PN model into a Sequential Function Chart (SFC) or ladder diagram language (LD)[2]. (Giua and DiCesare 1993; Uzam and Jones 1998).

Here, the SFC model is obtained by replacing each place of the PN model by a step. A control action is associated with each step that corresponds to the event (sensor) and belongs to the output transition. Transitions and events remain unchanged. This technique is inspired from the works presented in (Giua and DiCesare 1993; Uzam and Jones 1998). The SFC model for this example is presented in Figure 8.

[2]Sequential Function Chart (SFC) and Ladder Diagram (LD) are the PLC standard language that describe by IEC 1131-3 standard

Fig. 8. SFC model corresponding to the PN controller in Fig. 7

Actions A_1 and A_2 correspond to the assembly operations and action B_2 corresponds to the transfer operation. Sensors f_1, f_2 and t_2 detect the ends of operations.

6. Conclusion and future works

In this paper, we have presented a systematic method to reduce the number of linear constraints corresponding to the forbidden states for a safe PN. This is realized by using non- reachable states and by building the constraints using a systematic method. The important concept of over-state has been defined; it corresponds to a set of markings which keep the same property (forbidden or authorized). From the forbidden states, the set of over-states is calculated. The utilization of non-reachable markings allows great simplification of the constraints.

Properties which give necessary and sufficient conditions for the existence of a maximal permissive controller were established and illustrated for a manufacturing system. After the simplifications, the existence of the controller is proved. When this controller exists, the invariant approach allows the computation of the controller that can be transformed to a SFC model and be directly implemented in a PLC.

Our future work will include:

1) Developing this method of simplification to achieve more reduced results using the partial invariant idea,

2) Using this idea for simplification of conditions that are employed as predicates for controllable transitions. In this case, we can develop the idea of over-state for non-safe Petri Nets. The idea is to introduce the number of tokens as a power of the place identifiers. This can be indicated as follows: $P_1^3\ P_4^2$ …, place P_1 and place P_4 containing respectively 3 and 2 tokens. Thus, some of the properties presented in this paper can be generalized. Of course, some fundamental research needs to be done.

References

Basile F., Chiacchio P., Giua A., (2006), "Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places", *Automatica*, 42, 995-1004 .

Bratosin C., S.Caramihai, H.Alla. (2005). Synthesis of feedback control logic for safe Petri Nets. *The 15th international conference on control systems and computer science.* Bucharest, Romania

David R., Alla H., (2005) , *Discrete, Continuous, and Hybrid Petri Nets*, Springer, ch 1-3.

Dideban A., Alla H, (2005), "From forbidden state to linear constraints for the optimal supervisory control", *The 15th international conference on control systems and computer science*, 25 – 27 May, Bucharest, Romania

Ghaffari A., N. Rezg and X.-L. Xie, (2003b), "Design of Live and Maximally Permissive Petri Net Controller Using Theory of Regions", *IEEE Trans. On Robotics and Automation*, 19(1).

Giua A., F. DiCesare, M. Silva, (1992),"Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions", *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics* (Chicago, USA), pp. 974-799.

Giua A., F. DiCesare, (1993), "Grafcet and Petri Nets in Manufacturing ", in *Intelligent Manufacturing: Programming Environments for CIM*, W.A. Gruver and J.C. Boudreaux (Eds.), pp. 153-76, Springer-Verlag.
Giua A., Xie X., (2005), "Control of safe ordinary Petri nets using unfolding", *Discrete Event Dynamic Systems: Theory and Applications*, 15, 349–373.
Kumar R., Holloway L.E., (1996),"Supervisory control of deterministic Petri nets with regular specification languages", *IEEE Trans. Automatic Control*, 41(2):245-249.
Moody J. O., Antsaklis P., (2000), "Petri net supervisors for DES with uncontrollable and unobservable transition", *IEEE Trans. Automatic Control*, 45(3):462-476.
Morris Mano M., (2001), *Digital Design,* Prentice Hall , ch 3.
Ramadge P. J., and Wonham W. M., (1987), "Modular feedback logic for discrete event systems", *SIAM Journal of Control and Optimization*, 25 (5):1202-1218.
Ramadge P. J., and Wonham W., (1989), "The Control of Discrete Event Systems", Proceedings of the IEEE; *Special issue on Dynamics of Discrete Event Systems*, Vol. 77, No. 1:81-98.
Roussel J.-M., Giua A., (2005), "Designing dependable logic controllers using the supervisory control theory", 16th IFAC World Congress, CDROM paper n°04427, 6 pages, Praha(CZ), July 4-8.
Uzam M., Jones A. H., (1998), "Discrete event control System Design Using Automation Petri Nets and their Ladder Diagram Implementation" *Int J Adv Manuf Tech,* 14: 716-728.
Uzam M., (2002), "An Optimal Deadlock Prevention Policy for Flexible Manufacturing Systems Using Petri Net Models with resources and the Theory of Regions" *Int J Adv Manuf Tech,* 19: 192-208.
Yamalidou K., Moody J., Lemmon M.and Antsaklis P., (1996), "Feedback control of Petri Nets based on place invariants", *Automatica*, 32(1):15-28.
ZhiWu Li, MengChu Zhou, (2004), "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems" *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(1): 38-51.

APPENDIX I: ALGORITHMS

A) Algorithm 1: *Selection of the set of final over-states*

Step 1: Find the forbidden state M_i for which $Cv(M_i)$ (Definition 7) is: a) non null, b) the smallest one, and c) $Cf\ (M_i) = 0$;
If M_i does not exist, go to step 5;

Step 2: a) Find the set of constraints $C = \{c_1,\ldots, c_k,\ldots, c_m\}$ such that: $R(c_k,M_i) = 1$,
b) Find the constraint c_j in set C which covers the maximal number of states M_r. with $Cf\ (M_r) = 0$, and
c) Take the simpler c_j in case of equality.

Step 3: Save c_j in B_4;

Step 4: Mark the forbidden states which are covered by the constraint c_j in the line Cf; Go to step 1;

Step 5: End;

B) Algorithm 2 : *Complete algorithm for controller synthesis*

Step 1: Compute the set of over-states B_1 for the set of border forbidden state $\mathcal{M}_B$.

Step 2: Compute the set of over-states B_2 by deleting from B_1, the over states that exist in $\mathcal{M}_A$.

Step 3: Compute B_3 by deleting redundant over-states from B_2.

Step 4: Verifying Corollary 1 for maximal permissive controller: if it is verified go to step 5 else there is no maximal permissive controller. Go to Step 8.

Step 5: Apply algorithm 1 for computing B_4.

Step 6: Compute the control places from set of constraints B_4 by Yamalidou method.

Step 7: Transforming PN model into a SFC.

Step 8 : End

Abbas Dideban received his Ph.D. in Automation control from University of Grenoble I, France in 2007. He was awarded the M.Sc. degrees in Digital Electronic from Sharif University, Iran in 1997. He joined to the University of Semnan as a lecturer from 1998. At the same time he was cooperated with Jahad-daneshgahi Sharif in industrial Automation sector. Now he is an assistant professor at Semnan University. His research topics include Discrete Event Systems, Petri Nets, Industrial automation, Digital systems Design.

Hassane ALLA is Professor at the University Joseph Fourier of Grenoble. His research is mainly concerned with tools derived from Petri nets and automata used for the performance evaluation and for the control synthesis of discrete event systems. He is author or co-author of about one hundred publications. One of its main publications is a book on Continuous and Hybrid Petri nets which has been published in English and in French.