Time Floating General Mutual Exclusion Constraints (TFGMEC)

Zied Achour , Nidhal Rezg

LGIPM & INRIA/MACSI, Ile du Saulcy, 57045 Metz CEDEX, France

achour@loria.fr , nrezg@loria.fr

Abstract: In this paper, we propose a design methodology for Petri net controllers for the forbidden state-transition problem by taking into account the time constraints. The aim of the proposed approach is to solve a Forbidden state-transitions time depending and so called Time Floating General Mutual Exclusion Constraints. The proposed methodology is based on the Ramadge-Wonham approach and the theory of regions to generate a set of control places to add to the plant Petri net model.

Keywords: Discrete Event Systems, Supervisory Control, Timed place Petri net, GMEC.

Nidhal REZG is Professor at University Paul Verlaine-Metz since September 2004. He obtained his Ph.D. degree in industrial control systems from the Institut National des Sciences Appliquées de Lyon in 1996 and his ability to supervise research work from the University Paul Verlaine-Metz in 2003. Between 1997 and 1999 he was Professor in the Industrial Engineering Department at Monctoy University, New Brunswick, Canada. In September 1999 he obtained an associate professor position at University Paul Verlaine-Metz. His research activity deals with control synthesis of discrete dynamic event systems, the reliability and maintenance for performance evaluation simulation and optimisation of manufacturing systems. He was Guest Editor of a special issue of International Journal of Production Research in 2004.

Zied ACHOUR received his M.S. degree in Automatic production from the University Henri-Poincaré of Nancy-France in 2002 and his Ph.D. degree from the University Paul Verlaine-Metz, in 2005. Since September 2006 he is an Associate Professor at University Paul Verlaine-Metz. His research activity deals with control synthesis of discrete dynamic event systems. He is a member of I*PROMS project.

1. Introduction

The complexity of discrete event systems (DES) makes more difficult the realization of an effective and realistic control device. In this paper, bounded Petri nets are used for modelling DES, hereafter also referred to as plants. For a discrete model of a plant and a set of specifications, the goal of control synthesis is to determine a supervisor represented by a given set of legal markings which guarantees the desired behaviour. Supervisory control allows the avoidance of a set of forbidden states defined by some General Mutual Exclusion Constraints (GMEC) by adding some control places.

The pioneer work of Ramadge and Wonham [8] [9] proposed an automata-based framework for Discrete Event Systems control and addressed the existence and synthesis of the most permissive supervisor. Their approach is based on automata and formal language modeling. Unfortunately, the lack of structure of automata models limits the development of efficient control synthesis algorithms. To overcome these disadvantages, Petri nets (PN) were used to model plants to control and to design supervisors. Plant Petri net modes' transitions are fully observable and can be either controllable or uncontrollable. Structural and behavior properties of Petri net models are used to design efficient optimal real-time control policies for forbidden state problems of safe marked graphs [5], state machines [2], general Petri nets [6], and general marked graphs [3].

Consideration of time is crucial for realistic characterization of systems and hence leads to two types of concerns: the checking or validation of certain temporal specifications, and the control which requires temporal specifications. It is therefore important to integrate time in the specifications and synthesis. For this purpose, we introduce a new class of General Mutual Exclusion Constraints (GMEC) called Time Floating General Mutual Exclusion Constraints (TFGMEC), for which we propose a solution method inspired by a supervisor modelled by control places added to the initial Petri net model.

This paper is organized as follows. Section 2 presents the Time Floating General Mutual Exclusion Constraints. Section 4 is comprised of three subsections. In the first one, we give some terminology and definitions of the supervisory control problem. Theory of regions and the syntheses methodology are then introduced in subsections 3.2 and 3.3.

2. Time Floating General Mutual Exclusion Constraints (TFGMEC)

A Petri net is a tuple N= (P, T, Pre, $Post$, M), where P is a finite set of places, T is a finite set of transitions, with $P \cap T = \emptyset$, $Pre: P \times T \rightarrow \mathcal{N}$ and $Post: P \times T \rightarrow \mathcal{N}$ are respectively the pre-incidence function that defines weighted arcs from places to transitions and the post-incidence function that defines weighted arcs from transitions to places, where $\mathcal{N}$ is the set of non-negative integers and $M : P \rightarrow \mathcal{N}$ is the marking vector whose ith component, $M(p_i)$, is the number of tokens in the ith place. M_0 is an initial marking.

A timed place Petri net, or p-timed Petri net, has been adopted in this study. Time constants are associated with places; places are the steps or tasks that the project comprises. A token deposited in a place becomes available after that period of time (time execution of the task).

A p-timed Petri net is formally a bipartite directed graph: $TPN=(N, h)$, where

- N is a normal Petri net as previously defined,
- $h : P \rightarrow \mathcal{R}+$ is the place time function which represents the time delay associated with the relevant place.

The legal markings set corresponds to a linear inequalities set $wM \leq k$ called General Mutual Exclusion Constraints (GMEC), where $w \in Z^{n_c \times m}$, $k \in Z^{n_c}$, m is the number of places, n_c the number of constraints and M the current marking. These constraints can describe (generalized) mutual exclusion, deadlock prevention constraints, and others [1]. For example, we might wish to enforce the constraint $M(p_1)+M(p_2) \leq 1$, which means that at most one of the two places p_1 and p_2 can be marked, or, in other words, both places cannot be marked at the same time.

Time is an important factor to integrate to the plant model. Taking into account timed specifications makes the discrete model more realistic. This paper is concerned with timed specifications defined by GMEC. More precisely, a GMEC is considered in an interval of time and is not considered outside this interval. This kind of GMEC is called Time Floating General Mutual Exclusion Constraints (TFGMEC).

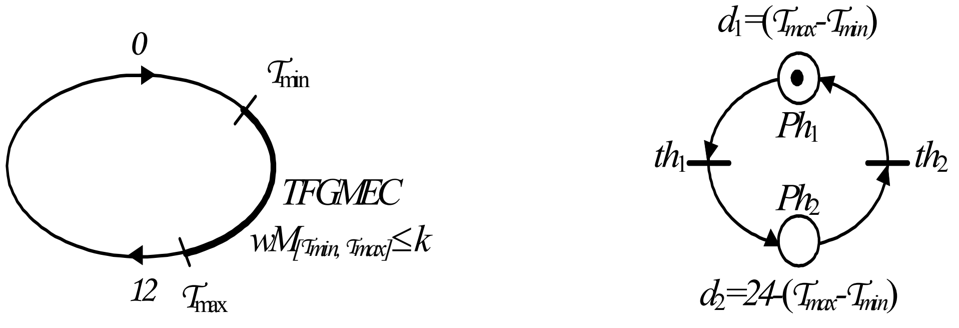
Definition 1. Time Floating General Mutual Exclusion Constraints are GMEC considered only in an interval of time $[\mathcal{T}_{\min}, \mathcal{T}_{\max}]$. TFGMEC are defined by a set of linear inequalities $wM_{[\mathcal{T}min,\ \mathcal{T}max]} \leq k$.

In the everyday life, many timed specifications can be defined by TFGMEC.

Examples: i) A production system sharing two resources in the interval of time [*8 a.m., 12 p.m.*] but only one outside this interval. ii) A plane which may not fly in a certain air lane in the interval of time [*11 p.m., 6 a.m.*].

Starting from a no-timed Petri net plan model and giving timed specifications, the goal of the control syntheses is to generate a set of control places to be added to the Petri net plan model. Control places, representing the supervisor, guarantee the desired behavior.

To solve TFGMEC problem (Figure 1a), the key idea consists of introducing into the plant model a global clock counting time. A P-timed Petri net is used for the global clock modelling (Figure 1b).


0
$\mathcal{T}_{\min}$
TFGMEC
$wM_{[\mathcal{T}min,\ \mathcal{T}max]} \leq k$
12
$\mathcal{T}_{\max}$
$d_1=(\mathcal{T}_{max}-\mathcal{T}_{min})$
Ph_1
th_1
th_2
Ph_2
$d_2=24-(\mathcal{T}_{max}-\mathcal{T}_{min})$

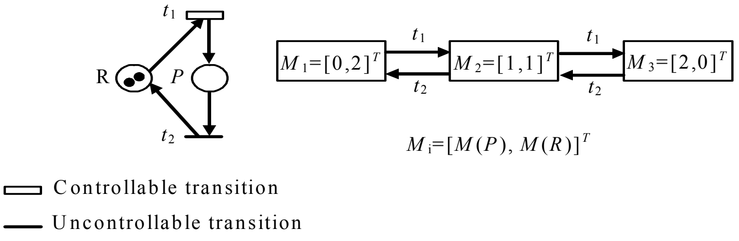
(*a*) *TFGMEC daily time cycle*

(*b*) *Global clock timed Petri net model*

Figure 1. Global Clock Modelling

According to the TFGMEC, the time cycle τ considered may be daily cycle, weekly cycle, etc. In the P-timed Petri net model of the global clock, d_i represents the time delay associated with the relevant place Ph_i. Note that $\sum d_i = \tau$.

Example 1: Consider the plant Petri net model and its reachability graph $RG(N, M_0)$ shown in Figure 2, which represents a production system sharing two resources R. For some noisiness reasons, one of the two resources must be stopped from *10 p.m.* until *7 a.m.* Formally, the TFGMEC which must be enforced is $M(p)_{[10\,p.m.,\ 7\,a.m.]} \leq 1$. In other words, place p may not contain more than one token in the interval of time [*10 p.m., 7 a.m.*].



(*a*) *Petri net model* (*b*) *Reachability graph*

Figure 2. Plant Model

In Figure 3, a Petri net N_1 model of the system (plant + global clock) depicted in Figure 2 is given. When a token arrives in the global clock place Ph_1 it becomes available after a period d_1 equal to nine hours, this being the period in which the TFGMEC is considered. In other words, a token in Ph_1 means that the current time is inside the interval [*10 p.m., 7a.m.*] while a token in Ph_2 means that the current time is outside this interval.

Assumption 1: A resource should not cease working until it has finished its task.

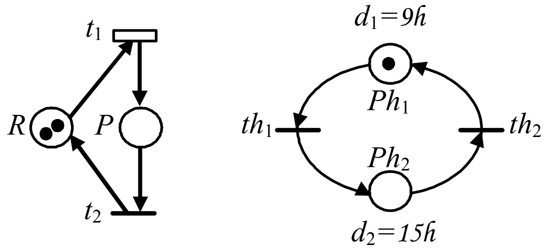


Figure 3. Petri Net Model of the Plant and Its Global Clock

Reachability graph $RG(N_1, M_0)$ of Figure 4 corresponds to the reachable states of the example of Figure 3. Transition t_1 fired from the marking M_2 leads to M_3 and violates the TFGMEC specification $M(p)_{[10\ p.m.,7\ a.m.]}\leq 1$. In fact, in M_3, the current time is inside the interval [*10 p.m., 7 a.m.*] ($M(Ph_1) = 1$) and two resources R are used ($M(P) = 2$). Instance $(M_2 \xrightarrow{t_1} M_3)$ is then a forbidden state transition. From Assumption 1, transition th_2 fired from the marking M_6 leads to M_3 too but it does not violate the TFGMEC specification.

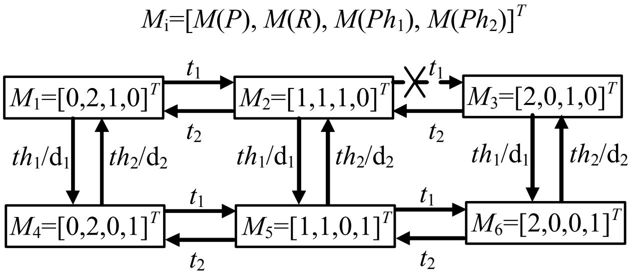


Figure 4. Desired Behavior

3. Supervisory Control Problem of TFGMEC

3.1. Terminology and definitions of supervisory control

Control specifications are given as a set of forbidden markings which correspond to undesirable states because they either compromise the system safety or they yield deadlock situations. Let M_F be the set of *forbidden* markings for which specifications do not hold.

Note that the set of transitions $T = T^c \cup T^u$ be partitioned into *controllable* transitions T^c and *uncontrollable* transitions T^u.

Supervisory control problem consists in synthesizing a supervisor that ensures the legal behaviour of the plant. Supervisor must by maximal permissive. With uncontrollable transitions, to respect the specifications, it is necessary to prevent the system from reaching a superset of the forbidden markings, containing all *dangerous* markings from which a forbidden one may be reached by firing a sequence of uncontrollable transitions.

Definition 2: A marking is said to be dangerous if it leads to a forbidden marking by firing uncontrollable transitions. Let M_D be the set of dangerous markings. Formally,

$M_D=\{M\in R(N,M_0)|\exists M'\in M_F\wedge\sigma\in (T^u)^*,M[\sigma>M'\}$. Obviously, $M_F \subseteq M_D$.

Liveness, or nonblockingness, requires the reachability of some marked states. In this paper, without loss of generality, we restrict our attention to the case where the initial marking state is the unique marked state. The liveness requirement is then equivalent to reversibility. Finally, the behavior of the controlled system under both safety specification and the liveness requirement can be defined as follows.

Definition 3: A marking is said blocking if it uncontrollably leads from markings in the Strongly Connected Component (SCC) containing M_0, to markings outside the SCC. Let M_B be the set of blocking markings. Formally $M_B=\{M\in SCC|\exists\ M'\notin SCC, \exists t\in T^u, M[t>M'\}$.

Definition 4: The set M_L of legal or admissible markings is the maximal set of reachable markings such that (i) $M_L\cap M_D = \varnothing$, (ii) it is possible to reach the marked marking M_0 from any legal marking without leaving the set M_L, (iii) any transition t from a legal marking to a non legal marking is a controllable transition. Let R_c be the reachability graph containing all legal markings.

We use the Ramadge-Wonham approach for the controlled behavior computation. Computation of the Strongly Connected Component (SCC) of controlled system is given by the following algorithm:

Algorithm 1: The controlled behavior computation

Step 1. *Generate the reachability graph $RG(N, M_0)$ of the plant Petri net model (N, M_0) to be controlled.*

Step 2. *Define the set of forbidden markings M_F.*

Step 3. *Identify the set M_D of dangerous markings by exploring $RG(N, M_0)$.*

Step 4. *Determine the controlled reachability graph R_c derived from $RG(N, M_0)$ by removing M_D.*

Step 5. *Verify that R_c is a SCC to enforce the liveness of the controlled system.*

Step 5.1. *If R_c is a SCC then go to step 8.*

Step 5.2. *If R_c is not a SCC then compute the SCC.*

Step 6. *Replace $RG(N, M_0)$ by the SCC determined in 5.2 and replace M_F by the set of markings outside SCC.*

Step 7. *Go to step 3.*

Step 8. *R_C is the legal behavior.*

Algorithm 1 gives the maximally permissive controller, since only the dangerous and blocking markings are eliminated. Let M_L be the set of all markings of R_c and Ω be the set of all state-transitions leading outside R_C. Formally, the set of state-transitions the controller has to disable is $\Omega=\{(M \xrightarrow{t} M')\ |M[t>M'\wedge M\in M_L\wedge\ M'\notin M_L\wedge t\in T^c\}$. Each element of Ω is denoted as *event separation instance*.

To summarize,

Definition 5: An optimal controller is the controller that ensures the reachability of all markings in M_L and that forbids all state-transitions in Ω.

Control requirements are expressed as a set of forbidden states or a set of forbidden state-transitions of $RG(N, M_0)$. Two classes of supervisory control problems exist, the forbidden states problem (FSP), and the forbidden state-transitions problem (FSTP) [7].

3.2. Control places design

Given the PN plant model (N,M_0) and the desired behavior R. In the following paragraphs, we present the design of control places using the theory of regions [4].

Consider any place p of the controlled net with initial marking $M_0(p)$ and incidence vector C(p, .). Note that p can be either a place of the plant model or a control place.

For any transition *t* from any marking *M* in R, i.e., *t* is the label of an outgoing arc of the node M in *R* :

$$M'(p) = M(p) + C(p,t),\ \forall\ M[t > M' \in R\ . \quad (1)$$

where M' is the new marking or equivalently the destination node of arc *t*.

Consider now any non-oriented cycle γ of the desired behavior R. Applying the state equation to nodes in γ and summing them up gives the following *cycle equation*:

$$\sum_{t \in T} C(p,t) \cdot \vec{\gamma}[t] = 0, \quad \forall \gamma \in S\ . \quad (2)$$

where $\vec{\gamma}[t]$ denotes the algebraic sum of all occurrences of *t* in γ with a weight 1 for each transition in any given direction of the cycle and a weight of -1 for a transition in the opposite direction and S is the set of non-oriented cycles of the graph. $\vec{\gamma}$ will be called the counting vector of γ.

Consider now each node M of the desired behavior R. According to the definition of both FSTP and FSP, there exists a non-oriented path Γ_M from the initial state M_0 to M. Applying equation (1) along the path leads to: $M(p) = M_0(p) + C(p,\cdot)\vec{\Gamma}_M$ where $\vec{\Gamma}_M$ is the counting vector of the path Γ_M defined similarly as $\vec{\gamma}$. There may exist several paths from M_0 to M. Under the cycle equations, the product $C(p,\cdot)\vec{\Gamma}_M$ is the same for all these paths. As a result, the path Γ_M can be arbitrarily chosen. The reachability of any marking *M* in *R* implies that:

$$M_0(p) + C(p,\cdot)\vec{\Gamma}_M \geq 0, \quad \forall M \in R. \quad (3)$$

which will be called the *reachability condition*.

Lemma 1: Any place of the plant model (N, M_0) or a control place of the controlled net satisfies both relations (2) and (3).

An impure control place p_c is a place for which there exists at least one transition that is both input and output transition of p_c. The self-loops are introduced to increase the control power of control places. With respect to any event separation instance ($M \xrightarrow{t} M'$) in Ω, only control places p_c with self-loop connecting p_c and t need to be considered and the goal is to forbid t from firing at M. As a result, for each event separation instance ($M \xrightarrow{t} M'$), we look for a control place p_c defined by its initial marking $M_0(p_c)$, its incidence vector $C(p_c, .)$ and the weight $C^{-}(p_c, t)$ of the arc connecting p_c to t.

First, according to Lemma 1, cycle equation (2) and reachability condition (3) hold. Since the control places are impure, the reachability conditions (3) are no longer enough to guarantee the reachability of markings M in R. Additional conditions are needed to ensure firability of a transition t enabled at any markings M in R:

$$M(p_c) = M_0(p_c) + C(p_c,\cdot)\vec{\Gamma}_M \geq C(p_c,t)^{-}, \quad \forall M[t > \text{in } R\ . \quad (4)$$

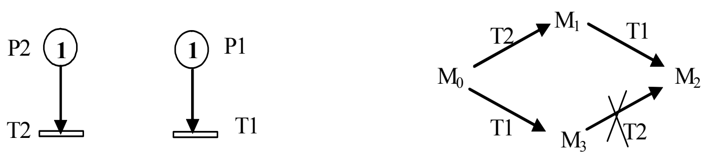
The event separation condition related to ($M^* \xrightarrow{t} M'$) becomes:

$$M_0(p_c) + C(p_c,\cdot)\vec{\Gamma}_{M^*} < C(p_c,t)^{-}. \quad (5)$$

Relations (2), (3), (4) and (5) are necessary and sufficient conditions for the existence of impure control places as it is stated by the following theorem.

Theorem 1:

A desired behavior R, subgraph of the reachability graph of a bounded Petri net (N,M_0), can be realized by adding impure control places to (N,M_0) iff there exists a solution ($M_0(p_c)$,$C(p_c,.,)$, C-(p_c, t)) satisfying conditions (2), (3) (4) and (5) for any element ($M \xrightarrow{t} M'$) of Ω.



(a) (b)

Figure 5. A Forbidden State-Transition Problem

Let us consider the problem of Figure 5. The control specification is to not fire T2 after T1. The event separation instance to solve is ($M_3 \xrightarrow{T2} M_2$) (see Figure 5.b.).The separation instance may be solved by a impure control place p_c iff p_c satisfies the following relations.

$M_0(p_c) \geq 0$

$M_0(p_c) + C(p_c,T1) \geq 0$

$M_0(p_c) + C(p_c,T2) \geq 0$

$M_0(p_c) + C(p_c,T1) + C(p_c,T2) \geq 0$

$M_0(p_c) \geq C^-(p_c,T2)$

$M_0(p_c) + C(p_c,T1) < C^-(p_c,T2)$

A control place defined by $M_0(p_c)$=1, $C(p_c,T1)$ = -1, $C(p_c,T2)$ = 0 and $C^-(p_c,T2)$ = 1 is a solution of the above system (see Figure 6).

Computation of an impure control place with respect to an event separation instance ($M^* \xrightarrow{t} M'$) can be performed in two steps. To understand it, note that combination of (4) and (5) leads to:

$$C(p_c,\cdot)\left(\vec{\Gamma}_M - \vec{\Gamma}_{M^*}\right) > 0, \quad \forall M[t> \text{ in } R \,. \tag{6}$$

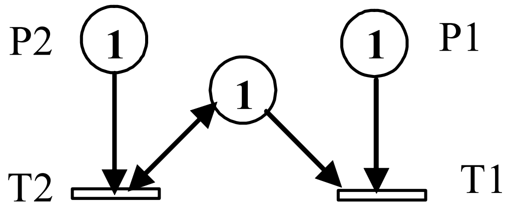


Figure 6. The Controlled Net Corresponding to Problem of Figure 5

Computation of the impure control place p_c first determines its incidence vector $C(p_c, .)$ by taking into account the cycle equations (2) and relations (6). The second step determines the initial marking and the weight $C^-(p_c,t)$ such that the place is a feasible solution. This can be achieved by setting:

$M_0(p_c)$=Max{0,Max{- $C(p_c, .)\,\vec{\Gamma}_M$: for all M in R}} and $C^-(p_c, t)$ = Min{$M_0(p_c) + C(p_c, .)\,\vec{\Gamma}_M$: for all M such that state-transition M[t > is in R}.

Corollary 1: There exists an impure control place solving an event separation instance ($M^* \xrightarrow{t} M'$) if the linear system defined by relations (2) and (6) has a solution.

3.3. Algorithm of the synthesis policy

Adding a set of control places to the plant model can optimally solve the supervisory control problem (if a solution of control exists). The aim of the supervision control is to disable the firing of a transition leading to an undesired marking. The algorithm of the synthesis method is as follows:

Algorithm 2: Synthesis methodology

Let a controlled reachability graph R_c derived from $RG(N, M_0)$ and a set Ω of event separation instances ($M \xrightarrow{t} M'$) with $M \in R_c$ be given.

Step 1. *Generate the reachability condition* (3) *for each marking M in R_c.*
Step 2. *Generate the cycle equations* (2) *related to the cycles in R_C.*

Step 3. *While* $\Omega \neq \varnothing$ *do*:

Step 3.1. *For any element* $(M_i \xrightarrow{t} M_j)$ *of* Ω *generate conditions* (*4*) *and* (*5*) *corresponding to each marking M in* R_c.

Step 3.2. *Solve the set of relations* (*2*), (*3*), (*4*) *and* (*5*). *Let* $(M_0(p_c), C(p_c, .), C^-(p_c, t))$ *be the solution if it exists. Otherwise, exit, as adding control places to the plant Petri net model cannot enforce the maximum permissive controlled behavior.*

Step 3.3. *Eliminate from* Ω *all the separation instances that can be solved by* $(M_0(p_c), C(p_c, .), C^-(p_c, t))$.

Step 4. *Remove redundant control places to obtain the controlled net by comparing the set of separation instances solved by each control places.*

Let us consider again the problem of Figure 2. The separation instance $(M_2 \xrightarrow{t_1} M_3)$ may be solved by an impure control place p_c if p_c satisfies the following relations:

Relation (2), the cycle equations:

$$C_c(p_c, t_1) + C_c(p_c, t_2) = 0$$
$$C_c(p_c, th_1) + C_c(p_c, th_2) = 0$$

Inequality (3), the reachability conditions:

$$M_{c0}(p_c) \geq 0;$$
$$M_{c0}(p_c) + C_c(p_c, t_1) \geq 0;$$
$$M_{c0}(p_c) + C_c(p_c, th_1) \geq 0;$$
$$M_{c0}(p_c) + C_c(p_c, th_1) + C_c(p_c, t_1) \geq 0;$$
$$M_{c0}(p_c) + C_c(p_c, th_1) + C_c(p_c, t_1) + C_c(p_c, t_1) \geq 0;$$
$$M_{c0}(p_c) + C_c(p_c, th_1) + C_c(p_c, t_1) + C_c(p_c, t_1) + C_c(p_c, th_2) \geq 0;$$

Inequality (4), to ensure firability of a transition t_1 enabled at the markings M_0, M_4, M_5, and finally, the separation condition (5):

$$M_{c0}(p_c) \geq C_c^-(p_c, t_1);$$
$$M_{c0}(p_c) + C_c(p_c, th_1) \geq C_c^-(p_c, t_1);$$
$$M_{c0}(p_c) + C_c(p_c, th_1) + C_c(p_c, t_1) \geq C_c^-(p_c, t_1);$$
$$M_{c0}(p_c) + C_c(p_c, t_1) < C_c^-(p_c, t_1).$$

Once relations (2), (3), (4) and (5) are established, we then determine $C(p_c, \cdot)$ and the initial marking of place p_c, which will make it possible to have the Petri net model of the controlled plant.

A solution may be given by $M_{c0}(p_c) = 2$, $C_c(p_c, t_1) = -1$, $C_c^-(p_c, t_1) = 2$, $C_c(p_c, t_2) = 1$, $C_c(p_c, th_1) = 1$ and $C_c(p_c, th_2) = -1$ (see Figure 7).

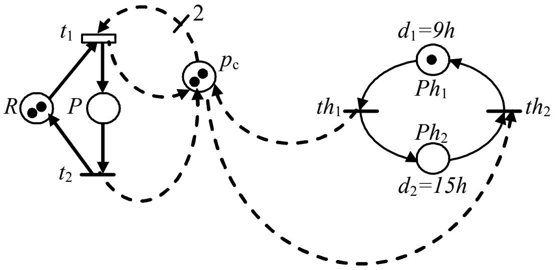


Figure 7. Controlled Net Corresponding to the Problem of Figure 2.

When a token becomes available in place ph_2, transition th_2 is fired since place p_c is always marked. The addition of the control place p_c does not perturb the global clock evolution.

Assumption 1, a resource should not cease working until it has finished its task means that if $M(p) = 2$ and then we are in the interval of time [*10 p.m., 7a.m.*], supervisor can not force the firing of transition t_2. Supervisor can only forbid the firing of transition t_1 in the interval of time [*10 p.m., 7a.m.*]. To respect the TFGMEC $M(p)_{[10\,p.m.,\ 7}$

$_{a.m.]}\leq 1$, the residence time of a token in the place p must be considered. In other terms, if we know that a token resides in the place p at the maximum one hour, then the TFGMEC can be $M(p)_{[9\,p.m.,\ 7\,a.m.]}\leq 1$.

Example 2: Consider a system of two jobs sharing two resources. Resources R1 and R2 are needed for performing jobs A and B. The PN plant model is given in Figure 8a. The set of controllable transitions is $T^c=\{t_1, t_4\}$. The reachability graph $RG(N, M_0)$ is given in Figure 4b.

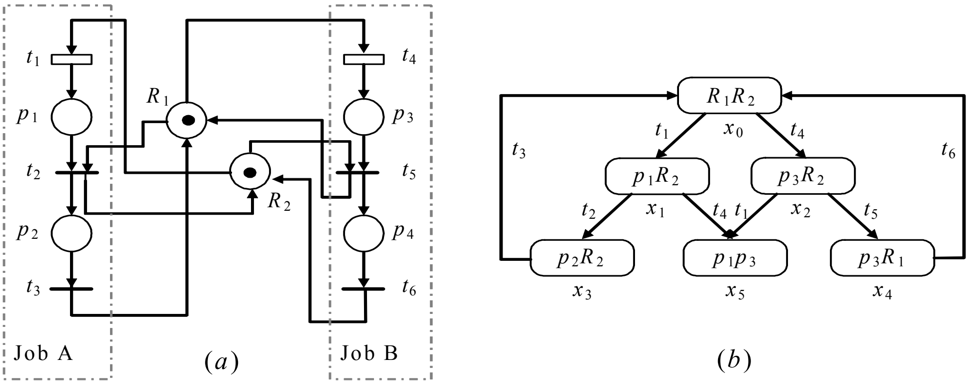


Figure 8. Production System

(a) Petri Net model (N, M_0)

(b) Reachability graph $RG(N, M_0)$

Plant Petri net model (N, M_0) of Figure 8a is not live since no transition can be fired from the state x_5. In this case, x_5= (1,0,1,0,0,0) is a blocking state. $\Omega=\{(x_1, t_4), (x_2, t_1)\}$ is the set of all state-transitions leading to x_5. Controlled reachability graph is given in Figure 9.

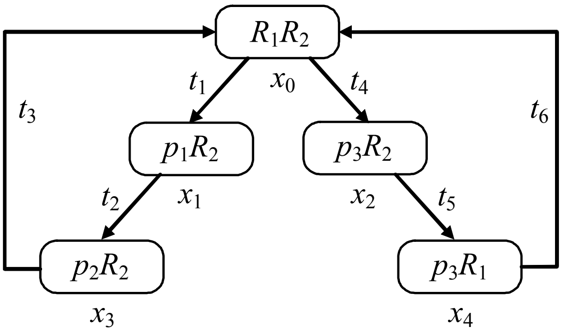


Figure 9. Controlled Reachability Graph

Let us use Algorithm 2 to synthesise the set of control places that will avoid the reachability of x_5. Only one separation condition related to the two separation instances needs to be considered, namely,

$$M_0(p_c)+C(p_c,t_1)+C(p_c,t_4)<0$$

The controlled reachability graph contains two different cycles and hence two cycle equations

$$C(p_c,t_4)+C(p_c,t_6)=0$$
$$C(p_c,t_1)+C(p_c,t_3)=0$$

whereas the reachability conditions can be expressed as follows:

$$y_0 :\rightarrow M_0(p_c)\geq 0$$
$$y_1 :\rightarrow M_0(p_c)+C(p_c,t_1)\geq 0$$
$$y_2 :\rightarrow M_0(p_c)+C(p_c,t_4)\geq 0$$

The above linear system has a solution $C(p_c, .\ ) = (-1, 0, 1, -1, 0, 1)$ and $M_0(p_c)=1$. The corresponding control place p_c is an input place of t_1 and t_4 and an output place of t_3 and t_6. The control place p_c contains initially one token. The controlled Petri Net is presented in Figure 10.

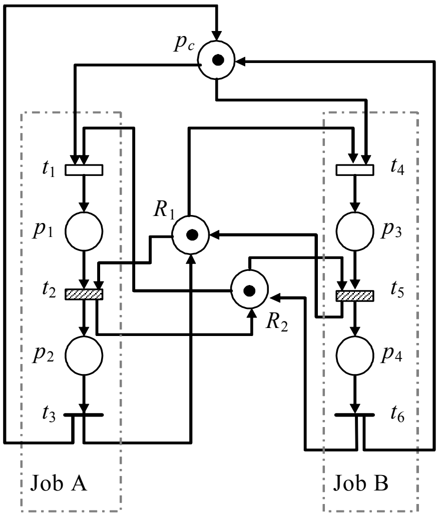


Figure 10. The Controlled Model of the System of Figure 4

Let us consider another kind of specifications. Suppose that only Job A must be used in the time interval [*8 a.m., 12 p.m.*], only Job B must be used in the time interval [*12 p.m., 6 p.m.*] and that both of them must be used outside these intervals. This kind of specification is called Time Floating General Mutual Exclusion Constraints (TFGMEC), which are considered next.

Consider again the production system given in Figure 8. The new specifications can be written as follow:

TFGMEC$_1$: only Job A must be used in [*8 a.m., 12 p.m.*]; $M(p_3)+M(p_4)_{[8\ a.m.,\ 12\ p.m.]} = 0$.

TFGMEC$_2$: only Job B must be used in [*12 p.m., 6 p.m.*]; $M(p_1)+M(p_2)_{[12\ p.m.,\ 6\ p.m.]} = 0$.

The timed Petri net global clock model relative to this example is given in Figure 11b. When the place Ph_1 is marked it means that we are in the interval of time [*8 a.m., 12 p.m.*]. A token deposited in this place becomes available after a period $d_1 = 4$. When the place Ph_2 is marked it means that we are in the interval of time [*12 p.m., 6 p.m.*]. A token deposited in this place becomes available after a period $d_2 = 6$. When the place Ph_3 is marked it means that we are in the interval of time [*6 p.m., 8 a.m.*]. A token deposited in this place becomes available after a period $d_3 = 14$.

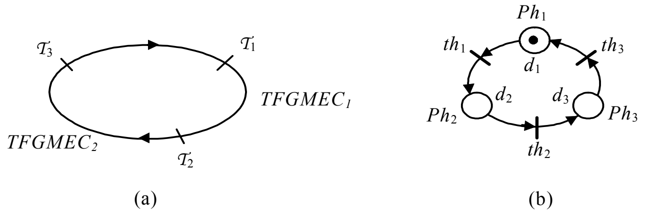


Figure 11. Global Clock Modelling Related to the Example Figure 4

(a) TFGMEC daily time cycle

(b) Timed Petri net model of the global clock

TFGMEC can be transformed as follow:

$M(p_3)+M(p_4)_{[08\ a.m.,\ 12\ p.m.]} = 0 \quad \Rightarrow \quad M(p_3)+M(p_4)+M(Ph_1) = 1.$

$M(p_1)+M(p_2)_{[12\ p.m.,\ 6\ p.m.]} = 0 \Rightarrow \quad M(p_1)+M(p_2)+M(Ph_2) = 1.$

Using the theory of regions, we obtain a supervisor represented by two control places p_{C2} and p_{C3} (Figure 12).

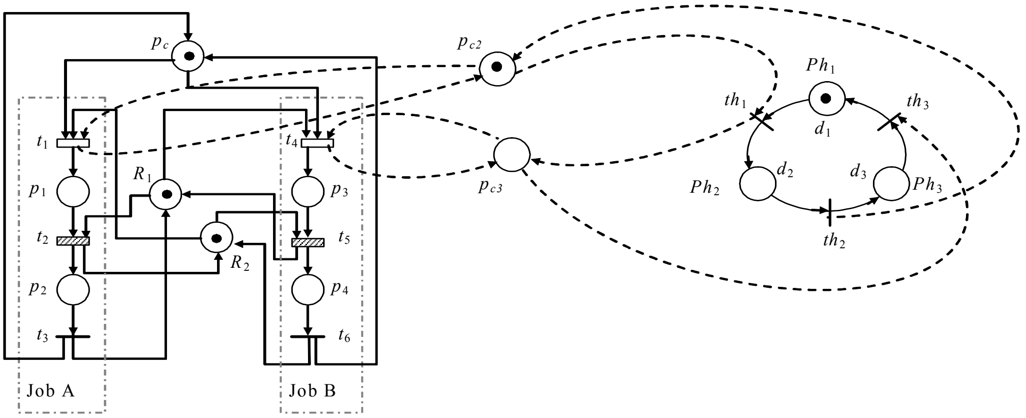


Figure 12. Controlled Model Taking into Account the TFGMEC

4. Conclusion

In this paper, we have proposed a design methodology for Petri net controllers for the forbidden state-transition problem. To solve control temporal specifications, a global clock is introduced in the plant Petri net model. Forbidden state-transitions are defined by some Time Floating General Mutual Exclusion Constraints. The proposed approach uses a Ramadge-Wonham approach to determine desired behavior and then uses the theory of regions to design a compiled Petri net controller that is a set of control places to add to the initial plant model.

REFERENCES

1. BADOUEL, E. and DAROUNDEAU P., **Theory of Regions**, Third Advanced Course on Petri Nets: Dagstuhl Castle, Vol. 1491 of Springer-Verlag Lecture Notes in Computer Science, pp. 529-586, 1998.

2. BOEL, R.K., BORDBAR B. and STREMERSCH G., **A min-plus polynomial approach to forbidden state control for general Petri nets**, Proc. WODES'98, Cagliari, Italy, pp. 79-84, 1998.

3. GHAFFARI, A., REZG N. and XIE X., **Feedback control logic for forbidden state problem of marked graphs**, IEEE Trans. on Automatic Control, Vol. 48/1, pp. 18-29, 2003.

4. GHAFFARI, A., REZG N. and XIE X., **Design of Live and Maximally Permissive Petri Net Controller Using Theory of Regions**, IEEE Trans. on Robotics and Automation, Vol. 19, No. 1, pp. 137-142, 2003.

5. HOLLOWAY, L. and KROGH B., **Synthesis of feedback control logic for a class of controlled Petri nets**, IEEE Trans. Automatic Control, Vol. 35, No. 5, pp. 514-523, 1990.

6. HOLLOWAY, L., GUAN X. and ZHANG L., **A generalization of state avoidance policies for controlled Petri nets**, IEEE Trans. Automatic and Control, Vol. 41, No. 6, pp. 804-816, 1996.

7. MOODY, J. O. and ANTSAKLIS P. J., **Supervisory Control of Discrete Event Systems Using Petri Nets**, Kluwer Academic Publishers, 1998.

8. RAMADGE, P.J. and WONHAM W.M., **Supervisory Control of a class discrete event processes**, SIAM Journal on Control and Optimization, Vol. 25, No. 1, pp. 206-230, 1987.

9. RAMADGE, P.J. and WONHAM W.M., **The Control of Discrete Event Systems**, Proceedings of IEEE, Vol. 77, pp. 81-98, 1989.