The 2002 American Control Conference, May 8-10, Anchorage, Alaska.

Synthesis of Supervisors Enforcing General Linear Vector Constraints in Petri Nets

Marian V. Iordache and Panos J. Antsaklis[1 2]

Abstract

This paper considers the problem of enforcing linear constraints containing marking terms, firing vector terms, and Parikh vector terms. Such constraints increase the expressivity power of the linear marking constraints. We show how this new type of constraints can be enforced in Petri nets. In the case of fully controllable and observable Petri nets, we give the construction of a supervisor enforcing such constraints. In the case of Petri nets with uncontrollable and/or unobservable transitions, we reduce the supervisor synthesis problem to enforcing linear marking constraints on a transformed Petri net.

1 Introduction

In this paper we consider a supervisory control problem for discrete event systems modeled as Petri nets, in which we desire to enforce a certain type of specifications. Thus we have a plant which is abstracted as a Petri net (PN), and a specification on the behavior of the PN plant. We desire to find a supervisor such that the closed-loop of the plant and the supervisor satisfies the specification. We restrict our attention to supervisors which can be represented as PNs, and to specifications in the form of conjunctions of linear inequalities involving the marking, the firing vector and the Parikh vector of the plant PN. We describe such specifications next.

Efficient methods have been proposed in [1, 5, 4, 7] for the synthesis of supervisors enforcing that the marking μ of a PN satisfies constraints of the form

$$L\mu \leq b \tag{1}$$

The methods address both the fully controllable and observable PNs and the PNs which may have uncontrollable and unobservable transitions. The constraints (1) have been extended in [4, 7] to the form

$$L\mu + Hq \leq b \tag{2}$$

[1]Department of Electrical Engineering, University of Notre Dame, IN 46556, USA. E-mail: iordache.1, antsaklis.1@nd.edu.

[2]The partial support of the National Science Foundation (NSF ECS99-12458), of DARPA/ITO-NEST Program (AF-F30602-01-2-0526) and of the Lockheed Martin Corporation is gratefully acknowledged.

which adds a firing vector term. In such constraints an element q_i of the firing vector q is set to 1 if the transition t_i is to be fired next; else $q_i = 0$. Without loss of generality, H has been assumed to have nonnegative elements. In this paper we consider constraints which add to (2) a Parikh vector term:

$$L\mu + Hq + Cv \leq b \tag{3}$$

In (3) v is the Parikh vector, that is v_i counts how often the transition t_i has fired since the initial marking μ_0. As an example, Parikh vector constraints can be used to describe fairness requirements, such as the constraint that the difference between the number of firings of two transitions is limited by one. Adding the Parikh vector term in (3) increases the expressivity power of linear constraints. In fact, any supervisor implemented as additional places connected to the transitions of a plant PN can be represented by constraints of the form

$$Hq + Cv \leq b \tag{4}$$

The contribution of this paper is as follows. In sections 2 and 3 we show that any place of a PN can be seen as a supervisor place enforcing a constraint of the form (4). Previously this property was known for constraints of the form $Cv \leq b$ and PNs without self-loops [3]. Then we show how to obtain supervisors enforcing constraints (3) in PNs. We first give the solution for the case of fully controllable and observable PNs in section 4. Then, in section 5 we turn our attention to PNs which may have uncontrollable and unobservable transitions. There we first define admissible constraints as the constraints for which the method for fully controllable and observable PNs can still be used. Then, by using net transformations, we reduce our problem to the supervisory synthesis problem for constraints of the form (1), for which effective methods exist. Our approach also extends the indirect method of [4] on enforcing constraints (2), as both coupled and uncoupled constraints can be considered. Finally, an example is given in section 6.

In the literature, Parikh vector constraints and marking constraints have been separately considered for vector DES (VDES) in [3]. The VDES considered in [3] correspond to PNs without self-loops. It has been shown there how to construct the optimal controller via integer programming. A less computationally burdensome approach, however not always optimal, has been

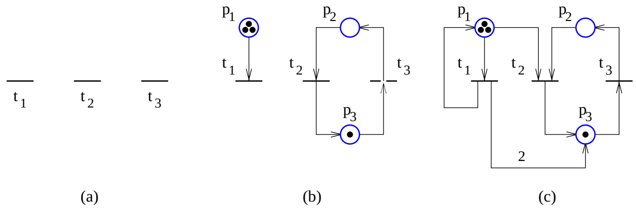

Figure 1: Petri nets for Example 2.1

given in [5, 4], which considers marking constraints and firing vector constraints. This paper extends some of the approaches of [5, 4] by including the Parikh vector constraints of [3].

2 Algebraic Representations of PNs

We denote a PN structure by $\mathcal{N} = (P, T, F, W)$, where P is the set of places, T the set of transitions, F the set of transition arcs, and W the weight function. We also denote by D the incidence matrix, and by D^+ and D^- its components corresponding to weights of arcs from transitions to places, and weights of arcs from places to transitions, respectively. The common algebraic PN representation is via the following state equation:

$$\mu = \mu_0 + Dv \tag{5}$$

where μ_0 is the initial marking. The operation of a PN can also be described through inequalities of the form (4). Indeed, from (5) we derive:

$$(-D)v \leq \mu_0 \tag{6}$$

Let $C = -D$. The inequality $Cv \leq \mu_0$ determines the operation of a PN only if the net has no self-loops. To deal with self-loops, an additional term is introduced:

$$Hq + Cv \leq \mu_0 \tag{7}$$

where

$$H_{i,j} = \begin{cases} D^+_{i,j} & \text{if } D^-_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Note that $H_{i,j} \geq 0$ for all i and j. The vector q has the following meaning. After we fire from μ_0 a sequence σ of Parikh vector v, the transition t_i is enabled iff $Hq^{(i)} + C(v + q^{(i)}) \leq \mu_0$, where $q^{(i)}$ is a vector q with zero elements except for the i'th one, which is one.

Example 2.1 Consider the PNs of Figure 1. The PN (a) is not restricted: the firings of t_1, t_2 and t_3 are free. Therefore H and C are empty matrices. However, by adding the places p_1, p_2 and p_3 as in the PN (b), the following inequalities appear in (7):

$$\begin{aligned} v_1 &\leq 3 & (9)\\ v_2 - v_3 &\leq 0 & (10)\\ -v_2 + v_3 &\leq 1 & (11) \end{aligned}$$

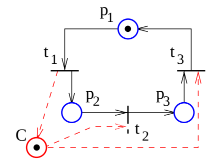

Figure 2: Illustrative example.

where the inequalities are generated by p_1, p_2, and p_3, respectively. The inequalities of the PN (c) are:

$$\begin{aligned} q_1 + v_2 &\leq 3 & (12)\\ v_2 - v_3 &\leq 0 & (13)\\ -2v_1 - v_2 + v_3 &\leq 1 & (14) \end{aligned}$$

Note that both μ and v can describe the state of a PN. We choose to denote by $\mathcal{R}(\mathcal{N}, \mu_0)$ all pairs (μ, v) such that $\mu_0 \xrightarrow{\sigma} \mu$, and the Parikh vector of the firing sequence σ is v.

3 Enforcing Generalized Linear Constraints

In this paper, a **supervisor** of a PN $\mathcal{N} = (P, T, F, W)$ is the PN implementation of a map $\Xi : \mathcal{M} \to 2^T$ for some[1] $\mathcal{M} \subseteq \mathbb{N}^{|P|} \times \mathbb{N}^{|T|}$. For simplicity, the supervisor is also denoted by Ξ. A supervisor Ξ restricts the operation of a Petri net $\mathcal{N}$ by forbidding all transitions $t \notin \Xi(\mu, v)$ to fire, where (μ, v) is the PN state. A PN $(\mathcal{N}, \mu_0)$ and a supervisor Ξ are in **closed-loop** if Ξ supervises $(\mathcal{N}, \mu_0)$; the closed-loop is denoted by $(\mathcal{N}, \mu_0, \Xi)$. Given $(\mathcal{N}, \mu_0, \Xi)$, we denote the set of all reachable states (μ, v) by $\mathcal{R}(\mathcal{N}, \mu_0, \Xi)$.

We desire to enforce constraints of the general form (3). Form (3) is more expressive than form (2). Indeed, consider the closed-loop PN of Figure 2. There is no place invariant involving the *control place* C, so C cannot be obtained by enforcing (2) [4]. However the following constraint of the form (3) describes C:

$$-v_1 + v_2 + v_3 \leq 1$$

In fact, as shown in the previous section, every place of a PN can be seen as a control place restricting the firings of the net transitions.

We say that a supervisor Ξ **enforces** (3) on a PN $(\mathcal{N}, \mu_0)$ if $\forall(\mu, v) \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$: (3) is satisfied. We say that Ξ **optimally enforces** (3) if $\forall(\mu, v) \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$: (a) Ξ is defined at (μ, v), and (b) a transition t_i enabled in the plant $(\mathcal{N}, \mu)$ is disabled by Ξ at (μ, v) (i.e. $t_i \notin \Xi(\mu, v)$) iff firing t_i leads to a state (μ', v') such that $L\mu' + Cv' \not\leq b$ or $L\mu + Hq^{(i)} + Cv \not\leq b$, where $q^{(i)}$ is the vector q corresponding to firing t_i.

[1] $|X|$ denotes the number of elements of X.

4 Supervisor synthesis in the fully controllable and observable case

This section describes the synthesis of the optimal supervisor enforcing constraints (3) in PNs in which all transitions are controllable and observable. The optimal supervisor is obtained by extending the formulas given in [6] for constraints of the form (2). Let

$$
\begin{aligned}
D_{lc}^{+} &= \max(0, -LD - C) &(15)\\
D_{lc}^{-} &= \max(0, LD + C) &(16)
\end{aligned}
$$

The supervisor is given by the incidence matrices:

$$
\begin{aligned}
D_{c}^{+} &= D_{lc}^{+} + \max(0, H - D_{lc}^{-}) &(17)\\
D_{c}^{-} &= \max(D_{lc}^{-}, H) &(18)
\end{aligned}
$$

The initial marking of the supervisor is:

$$\mu_{c0} = b - L\mu_0 \qquad (19)$$

where μ_0 is the initial marking of the plant. Note that in equations (15−18) the operator max is defined as follows. If A is a matrix, $B = \max(0, A)$ is the matrix of elements $B_{ij} = 0$ for $A_{ij} < 0$, and $B_{ij} = A_{ij}$ for $A_{ij} \geq 0$. Furthermore, for two matrices A and B of the same size, $C = \max(A, B)$ is the matrix of elements $C_{ij} = \max(A_{ij}, B_{ij})$.

Note that equations (17), (18) and (19) define a supervisor which can be represented as a PN of incidence matrices D_c^+ and D_c^-, and with initial marking μ_{c0}. We call the places of the supervisor **control places**.

Theorem 4.1 *The supervisor defined by the incidence matrices D_c^+ and D_c^- of (17) and (18) and of initial marking given by (19), optimally enforces (3).*

The theorem can be proved by verifying that in the closed-loop net (which has the incidence matrices $[D^{+T}, D_c^{+T}]^T$, $[D^{-T}, D_c^{-T}]^T$ and the initial marking $[\mu_0^T, \mu_{c0}^T]^T$), a control place prevents a transition t to fire iff firing t violates (3). To this end it can be proven by induction that

$$\mu_c = b - Cv - L\mu \qquad (20)$$

Note that the supervisors we build for (3) may not create a place invariant in the closed-loop net.

5 Supervisor synthesis in the case of PNs with uncontrollable and/or unobservable transitions

5.1 Admissibility

A transition is uncontrollable if the supervisors are not given the ability to directly inhibit it. A transition is unobservable if the supervisors are not given the ability to directly detect its firing. In our paradigm the supervisors observe transition firings, not markings. For instance, consider the PN of Figure 3. Assume first that t_1 is controllable and t_2 is uncontrollable. Then, in case (a) t_2 cannot be directly inhibited; it will eventually fire. However, in case (b) t_2 can be indirectly prevented to fire by inhibiting t_1. Now assume that t_2 is unobservable and t_3 is observable. This means that we cannot detect when t_2 fires. In other words, the state of a supervisor is not changed by firing t_2. However, we can indirectly detect that t_2 has fired by detecting the firing of t_3.

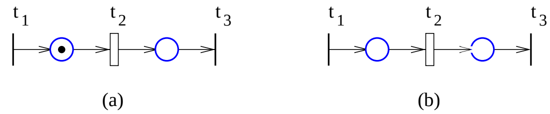

Figure 3: Uncontrollability/unobservability illustration.

We are interested in *admissible* constraints, that is constraints which can be optimally enforced as in section 4, in spite of our inability to detect or control certain transitions. We formally define admissibility as follows.

Definition 5.1 *Let $(\mathcal{N}, \mu_0)$ be a PN. Assume that we desire to enforce a set of constraints (3). Consider the supervisor defined by (17), (18), and (19). We say that the constraints (3) are* **admissible** *if for all reachable states (μ, v) of the closed-loop net it is true that:*

1. *If t is uncontrollable and t is enabled by*[2] *$\mu|_{\mathcal{N}}$ in $\mathcal{N}$, then t is enabled by μ in the closed-loop net.*

2. *If t is unobservable and t is enabled by μ, firing t does not change the marking of the control places.*

Note that condition 2 in the definition corresponds to the requirement that the unobservable transitions which are not dead at the initial marking of the closed-loop net, have null columns in $D_c = D_c^+ - D_c^-$ (where D_c^+ and D_c^- are defined in (17) and (18)). For general PNs it may not be easy to check whether a constraint is admissible. A computationally simple test is given in the following sufficient condition. Let $D_{c,uc}^-$ be the restriction of D_c^- to the columns of the uncontrollable transitions. Let $D_{c,uo}$ be the restriction of D_c to the columns of the unobservable transitions.

Proposition 5.1 *The constraints (3) are admissible at all initial markings if $D_{c,uo}$ and $D_{c,uc}^-$ are null matrices.*

The condition $D_{c,uo} = 0$ ensures that for any uncontrollable transition, a control place is either not connected to it, or is connected to it with input and output arcs of equal weight. The condition $D_{c,uc}^- = 0$ ensures that no control place is in the preset of an uncontrollable transition.

[2] We denote by $\mu|_{\mathcal{N}}$ the restriction of μ to the places of $\mathcal{N}$.

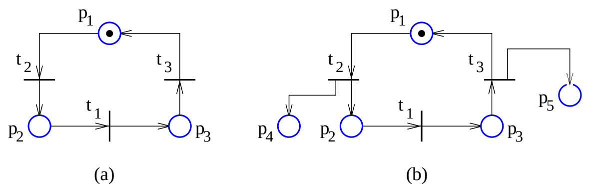

Figure 4: Illustration of the C-transformation.

5.2 Transformations to admissible constraints

When a constraint is admissible, it can be enforced as in section 4. However, when a constraint is not admissible or we cannot discern whether it is admissible, we are interested to transform it to a form which we know is admissible. Thus we have the following problem. Given a set of constraints (3) on a PN $(\mathcal{N}, \mu_0)$, find a set of admissible constraints

$$L_a\mu + H_a q + C_a v \leq b_a \tag{21}$$

so that if Ξ is a supervisor optimally enforcing (21) on $(\mathcal{N}, \mu_0)$, then $\forall(\mu, v) \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$: (3) is satisfied.

In section 5.5 we consider a transformation approach in which we transform the PN such that the constraints (3) are mapped into marking constraints. Then the marking constraints can be transformed to admissible constraints by using any of the approaches in [5]. First we define the PN transformations we use.

5.3 The C-Transformation

We illustrate the idea of the transformation on an example. Assume that we desire to enforce the constraint below on the PN of Figure 4(a)

$$\mu_1 + q_1 + v_2 - v_3 \leq 3 \tag{22}$$

By transforming the net as in Figure 4(b), (22) can be written without referring to v:

$$\mu_1 + q_1 + \mu_4 - \mu_5 \leq 3 \tag{23}$$

We say that the PN of Figure 4(b) and the constraint (23) are the C-transformation of the PN of Figure 4(a) and of (22).

The inverse C-transformation is also possible. Given the constraint

$$\mu_1 - 3\mu_4 + 2\mu_5 + q_1 \leq 5 \tag{24}$$

on the PN of Figure 4(b), we can map it to

$$\mu_1 + q_1 - 3v_2 + 2v_3 \leq 5 \tag{25}$$

in the original PN. We proceed next to formally define the direct and inverse transformations.

The C-Transformation

Input: The PN $\mathcal{N} = (P, T, F, W)$, the constraints $L\mu + Hq + Cv \leq b$, and optionally the initial marking μ_0.

Output: The C-transformed PN $\mathcal{N}_C = (P_C, T, F_C, W_C)$, the C-transformed constraint $L_C\mu_C + Hq \leq b$, and the initial marking μ_{0C} of $\mathcal{N}_C$.

1. Initialize $\mathcal{N}_C$ to equal $\mathcal{N}$, L_C to L, and let $k = |P|$.

2. For $i = 1$ to $|T|$

2.a. If C_i, the i'th column of C, is not zero

2.a.i. Set $k = k + 1$

2.a.ii. Add a new place p_k to $\mathcal{N}_C$ such that $p_k\bullet = \emptyset$ and $\bullet p_k = \{t_i\}$.

2.a.iii. Set $L_C = [L_C, C_i]$ and $\mu_{0C} = [\mu_{0C}^T, 0]^T$.

The C^{-1}-Transformation

Input: The PN $\mathcal{N} = (P, T, F, W)$, the C-transformed net $\mathcal{N}_C = (P_C, T, F_C, W_C)$, and a set of constraints $L_C\mu_C + Hq \leq b$ on $\mathcal{N}_C$.

Output: The constraints $L\mu + Hq + Cv \leq b$.

1. Set L to L_C restricted to the first $|P|$ columns and C to be a null matrix.

2. For $i = |P| + 1$ to $|P_C|$

2.a. Let j be the transition index such that $\bullet p_i = \{t_j\}$.

2.b. Set $C_j = L_{C,i}$. [3]

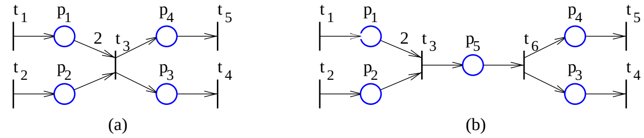

Figure 5: Example for the H-transformation.

5.4 The H-transformation

This transformation is a modification of the indirect method for enforcing firing vector constraints in [5]. We illustrate it on an example. Consider the PN of Figure 5(a). Assume that we desire to enforce

$$\mu_1 + \mu_2 + 2\mu_3 + q_3 \leq 5 \tag{26}$$

Then we transform the PN as shown in Figure 5(b). The transformation adds a place and a transition which correspond to the factor q_3. The transformed constraint is

$$\mu_1 + \mu_2 + 2\mu_3 + 4\mu_5 \leq 5 \tag{27}$$

where the term $4\mu_5$ is obtained as follows. Consider firing t_3 in the transformed net. If $\mu \xrightarrow{t_3} \mu'$ and a is the coefficient of μ_5, we desire

$$a + \mu_1' + \mu_2' + 2\mu_3' = 1 + \mu_1 + \mu_2 + 2\mu_3$$

where the factor 1 is the coefficient of q_3 in (26). Thus we obtain $a = 4$.

Next we formally define the H-transformation.

[3] $C_j/L_{C,i}$ is the column j/i of C/L_C.

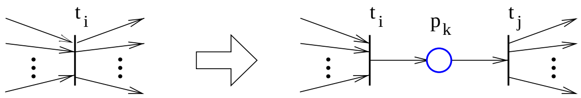

Figure 6: Illustration of the transition split operation.

The H-Transformation

Input: The PN $\mathcal{N} = (P, T, F, W)$, the constraints $L\mu + Hq \leq b$, and optionally the initial marking μ_0.

Output: The H-transformed PN $\mathcal{N}_H = (P_H, T_H, F_H, W_H)$, the H-transformed constraint $L_H\mu_H \leq b$, and the initial marking μ_{0H} of $\mathcal{N}_H$.

1. Initialize $\mathcal{N}_H$ to equal $\mathcal{N}$, L_H to L, and let $j = |T|$ and $k = |P|$.

2. For $i = 1$ to $|T|$

2.a. If H_i, the i'th column of H, is not zero

2.a.i. Set $j = j + 1$ and $k = k + 1$.

2.a.ii. Add a new place p_k and a new transition t_j to $\mathcal{N}_H$ as in Figure 6, where t_j has the same controllability and observability attributes as t_i.

2.a.iii. Set $L_H = [L_H,\ H_i + LD_i^-]$ and $\mu_{0H} = [\mu_{0H}^T,\ 0]^T$, where D_i^- is the i'th column of D^-, and D^- corresponds to $\mathcal{N}$.

The H^{-1}-Transformation

Input: The PN $\mathcal{N} = (P, T, F, W)$, the H-transformed net $\mathcal{N}_H = (P_H, T_H, F_H, W_H)$, and a set of constraints $L_H\mu_H \leq b$ on $\mathcal{N}_H$.

Output: The constraints $L\mu + Hq \leq b$.

1. Set L to L_H restricted to the first $|P|$ columns and H to be a null matrix.

2. For $k = |P| + 1$ to $|P_H|$

2.a. Let i be the transition index such that $\bullet p_k = \{t_i\}$.

2.b. Set $H_i = L_{H,k} - L_H D_{H,i}^-$. [4]

5.5 Algorithm for the transformation to admissible constraints

We can use the C- and H-transformations to obtain admissible constraints as follows.

Input: A PN $\mathcal{N}$, constraints $L\mu + Hq + Cv \leq b$, and optionally[5] an initial marking μ_0.

Output: Admissible constraints $L_a\mu + H_a q + C_a v \leq b_a$

1. Initialize L_a to L, H_a to H, and C_a to C.

2. Apply the C-transformation. Let $\mathcal{N}_C$, $L_C\mu_C + Hq \leq b$, and μ_{0C} be the C-transformed net, the constraints, and the initial marking, respectively.

3. Apply the H-transformation to $\mathcal{N}_C$, $L_C\mu_C + Hq \leq b$, and μ_{0C}. Let $\mathcal{N}_{HC}$, $L_{HC}\mu_{HC} \leq b$, and μ_{HC0} be the H-transformed net, the constraints, and the initial marking, respectively.

4. Test whether $L_{HC}\mu_{HC} \leq b$ is admissible. If so, exit, and declare $L\mu + Hq + Cv \leq b$ admissible.

5. Transform $L_{HC}\mu_{HC} \leq b$ to admissible constraints $L_{HCa}\mu_{HC} \leq b_a$, such that a supervisor optimally enforcing $L_{HCa}\mu_{HC} \leq b_a$ also enforces $L_{HC}\mu_{HC} \leq b$.[6] In case of failure, exit and declare failure to find admissible constraints.

6. Apply the H^{-1}-transformation to $L_{HCa}\mu_{HC} \leq b_a$. Let $L_{Ca}\mu_C + H_a q \leq b_a$ be the transformed constraint.

7. Apply the C^{-1}-transformation to $L_{Ca}\mu_C + H_a q \leq b_a$. Set $L_a\mu + H_a q + C_a v \leq b_a$ to the C^{-1}-transformed constraints.

We prove the following result in [2].

Theorem 5.1 *Assume that the algorithm does not fail at step 5. Then $L_a\mu + H_a q + C_a v \leq b_a$ is admissible, and a supervisor optimally enforcing it enforces also $L\mu + Hq + Cv \leq b$.*

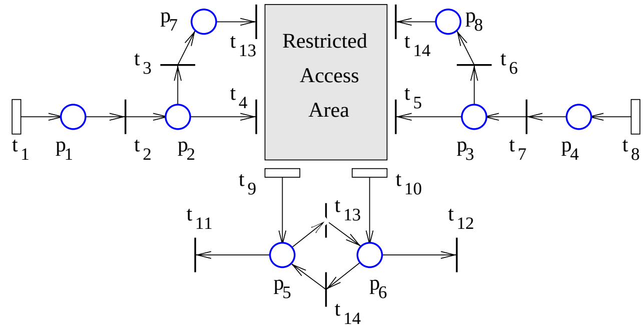

Figure 7: Plant Petri net in the example.

6 Example

Consider the plant PN of Figure 7. It corresponds to a region of a factory cell in which autonomous vehicles (AV) access a restricted area (RA). The number of AVs which may be at the same time in the RA is limited. The AVs enter the RA from two directions: left and right; AVs coming on the left side enter via t_4 or t_{13}, and AVs coming on the right side via t_5 or t_{14}. The AVs exit the restricted area via t_9 or t_{10}. The total marking of p_1, p_2 and p_7 corresponds to the number of left AVs waiting in line to enter the RA; only one AV should be in the states p_2 and p_7, that is $\mu_2 + \mu_7 \leq 1$.

[4] $H_i/L_{H,k}/D_{H,i}^-$ is the column $i/k/i$ of $H/L_H/D_H^-$, and D_H^- corresponds to $\mathcal{N}_H$.

[5] It is possible to carry out the algorithm independently of the initial marking.

[6] Any of the approaches in [5, 4] can be used. Approaches generating disjunctive constraints can also be used by applying the steps 6 and 7 to each component of the disjunction.

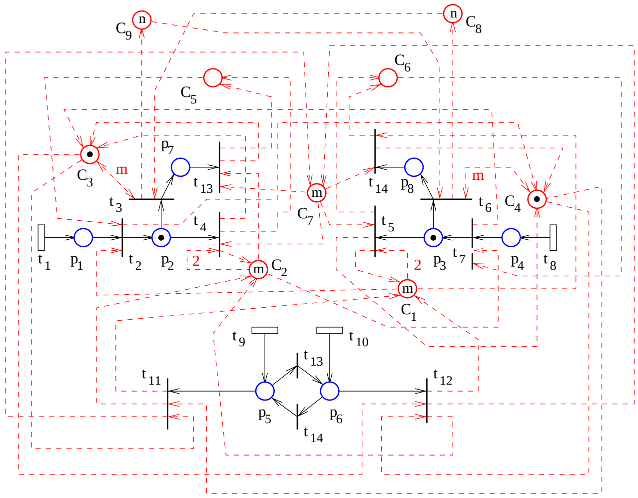

Figure 8: Closed-loop Petri net.

The marking of p_3, p_4, and p_8 has a similar meaning.

Let m be the maximum number of AVs which can be at the same time in the RA; note that the number of AVs in the RA is $v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10}$. When the number of vehicles in the restricted area is $m - 1$ and both a left and a right AV attempt to enter the restricted area (i.e. both $\mu_2 + \mu_7 = 1$ and $\mu_3 + \mu_8 = 1$), arbitration is required. When an AV is in p_2 and no arbitration is required, it can enter the RA without stopping. When arbitration is required, it stops (enters the state p_7) and waits the arbitration result. The same apply to p_3 and p_8. We desire the following. When an AV enters the RA, if an arbitration was required to decide that it may enter, the AV should enter via t_{13} or t_{14}; if no arbitration was required, it should enter via t_4 or t_5. These constraints can be written as follows:

$$2q_5 + \mu_2 + \mu_7 \leq m - (v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10}) + 1 \quad (28)$$

$$2q_4 + \mu_3 + \mu_8 \leq m - (v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10}) + 1 \quad (29)$$

$$mq_3 \leq \mu_3 + \mu_8 + v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10} \quad (30)$$

$$mq_6 \leq \mu_2 + \mu_7 + v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10} \quad (31)$$

In addition we have the requirements that

$$\mu_2 + \mu_7 \leq 1 \quad (32)$$

$$\mu_3 + \mu_8 \leq 1 \quad (33)$$

The requirement on the maximum number of AVs in the RA is

$$v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10} \leq m \quad (34)$$

We add the fairness constraints

$$v_3 - v_6 \leq n \quad (35)$$

$$-v_3 + v_6 \leq n \quad (36)$$

As t_1, t_8, t_9, t_{10} are uncontrollable and t_9, t_{10} unobservable, the constraints (28−31) and (34) are inadmissible. They are transformed to[7]

[7]The constraints (30) and (31) cannot be transformed to (more restrictive) admissible constraints; (39) and (40) represent relaxed (and admissible) forms of (30) and (31).

$$\begin{aligned} 2q_5 + \mu_2 + \mu_5 + \mu_6 + \mu_7 + v_{13} & \\ +v_{14} + v_4 + v_5 - v_9 - v_{10} &\leq m + 1 \quad (37) \\ 2q_4 + \mu_3 + \mu_5 + \mu_6 + \mu_8 + v_{13} & \\ +v_{14} + v_4 + v_5 - v_9 - v_{10} &\leq m + 1 \quad (38) \\ mq_3 - \mu_3 - \mu_8 - \mu_5 - \mu_6 - (v_{13} & \\ +v_{14} + v_4 + v_5 - v_9 - v_{10}) &\leq 0 \quad (39) \\ mq_6 - \mu_2 - \mu_7 - \mu_5 - \mu_6 - (v_{13} & \\ +v_{14} + v_4 + v_5 - v_9 - v_{10}) &\leq 0 \quad (40) \\ v_{13} + v_{14} + v_4 + v_5 - v_9 - v_{10} & \\ +\mu_5 + \mu_6 &\leq m \quad (41) \end{aligned}$$

The closed-loop PN is shown next to the plant in Figure 8, where the control places $C_1 \ldots C_9$ correspond to the constraints (37), (38), (39), (40), (32), (33), (41), (35), and (36), in this order.

7 Conclusion

Enforcing linear marking and firing vector constraints can be done effectively in Petri nets. This paper has extended this class of constraints to include Parikh vector constraints. Then, we have shown how these more expressive constraints can be enforced as effectively as linear marking constraints. We have also enhanced the previous technique for enforcing firing vector constraints in the presence of uncontrollable and unobservable transitions. Our algorithms are software implemented.

References

[1] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 974–979, October 1992.

[2] M. V. Iordache and P. J. Antsaklis. Synthesis of supervisors enforcing general linear vector constraints in Petri nets. Technical report isis-2002-002, University of Notre Dame, February 2002.

[3] Y. Li and W. Wonham. Control of Vector Discrete-Event Systems II - Controller Synthesis. *IEEE Transactions on Automatic Control*, 39(3):512–530, March 1994.

[4] J. O. Moody and P. J. Antsaklis. *Supervisory Control of Discrete Event Systems Using Petri Nets.* Kluwer Academic Publishers, 1998.

[5] J. O. Moody and P. J. Antsaklis. Petri net supervisors for DES with uncontrollable and unobservable transitions. *IEEE Transactions on Automatic Control*, 45(3):462–476, March 2000.

[6] J. O. Moody, M. V. Iordache, and P. J. Antsaklis. Enforcement of event-based supervisory constraints using state-based methods. In *Proceedings of the 38th Conference on Decision and Control*, pages 1743–1748, 1999.

[7] E. Yamalidou, J. O. Moody, P. J. Antsaklis, and M. D. Lemmon. Feedback control of Petri nets based on place invariants. *Automatica*, 32(1):15–28, January 1996.