Proceedings of the 38th
Conference on Decision & Control
Phoenix, Arizona USA • December 1999

WeM01 14:20

Enforcement of Event-Based Supervisory Constraints Using State-Based Methods

John O. Moody
Lockheed Martin Federal Systems
1801 State Rt. 17C, MD 0210
Owego, NY 13827-3998
john.moody@lmco.com

Marian V. Iordache and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame,
Notre Dame, IN 46556
miordach@maddog.ee.nd.edu, Panos.J.Antsaklis.1@nd.edu

Abstract

Efficient, established techniques exist for enforcing systems of linear inequalities on the state vectors of discrete event systems modeled as Petri nets. This paper presents methods for extending the convenience of these state-based techniques to supervisory constraints that are expressed in terms of allowable plant events rather than just allowable plant states. Two interpretations of event-based constraints are described. The direct interpretation assumes that the controller will actively disable events that would directly violate the constraints. The indirect interpretation indicates that the controller should prevent those states that could lead to the violation of the constraints. Methods for automatically synthesizing supervisors that use either of these interpretations are presented here along with mathematical results for integrating these methods with established techniques for handling uncontrollable and unobservable plant transitions. The technique is illustrated with a process control example.

1 Introduction

Established techniques [1, 7] exist for enforcing linear inequalities on the state vectors (or marking behavior) of Petri nets [6]. These techniques yield efficiently computed supervisors and incorporate methods for handling uncontrollable or unobservable transitions [3–5]. The constraints themselves have the form

$$l^T \mu_p \leq b \tag{1}$$

where $\mu_p \in \mathbb{Z}^n, \mu_p \geq 0$ is the marking vector of the plant, $l \in \mathbb{Z}^n$, $b \in \mathbb{Z}$, and $\mathbb{Z}$ is the set of integers.

Supervisory control goals may involve the firing vector of the Petri net as well as or opposed to the places. For example one might need to insure that two transitions do not fire simultaneously or that a certain transition is never allowed to fire when a certain place holds a token. There are two ways that constraints like these may be viewed. For the constraint

$$\mu_i + q_j \leq 1 \tag{2}$$

where μ_i is an element of the Petri net's marking vector and q_j is an element of its firing vector, do we mean that *transition j should be disabled whenever place i contains a token*, or do we mean that all plant states that would allow transition j to be enabled are forbidden whenever place i contains a token? The answer to this question lies in the particulars of a given plant and its operation. Both means of enforcing the constraint can be useful for different problems.

Uncontrollable and unobservable transitions are handled by transforming inadmissible constraints into *admissible constraints* [5], this concept is reviewed in section 2. The primary contribution of this paper is the extension of these results for synthesizing controllers that enforce constraints on both the marking and the firing vector. Section 3 describes rules for enforcing these constraints using the "direct" interpretation, i.e., transitions are explicitly disabled in order to satisfy the inequality. Algebraic schemes for handling the "indirect" interpretation of firing vector constraints were proposed in [7]. A new approach is presented in section 4 that uses the concept of uncontrollable transitions to force a correct interpretation of each constraint, thus avoiding the enumeration of separate cases that appeared in [7]. Section 5 illustrates the direct approach through a process control example involving the management of fluid levels in three tanks. Concluding remarks appear in section 6.

2 Admissible and Inadmissible Constraints

Given a constraint, $l^T \mu_p \leq b$, a supervisor must work to insure that the constraint is never violated directly and may never be violated through the firing of uncontrollable transitions or through incomplete knowledge due to unobservable transitions. In order to avoid expensive online searches by the supervisor through the uncontrollably reachable markings of the plant, the approach taken in [4,5] is to actually modify the constraints themselves such that the new constraints account for uncontrollability and unobservability. The following definitions are useful in understanding the motivation for the transformation of constraints. Unobservable transitions are also assumed to be uncontrollable. See [5]

[1]The partial financial support of the National Science Foundation (ECS95-31485) and the Army Research Office (DAAG55-98-1-0199) is gratefully acknowledged.

for greater detail on admissible and inadmissible constraints and the proofs for the items below.

Definition 1. **An admissible marking** μ_p is a marking such that

1. $l^T \mu_p \leq b$, and

2. For all markings μ'_p reachable from μ_p through the firing of uncontrollable transitions, $L\mu'_p \leq b$.

If either of these conditions is not met, then the marking is **inadmissible.**

Definition 2. Given a plant with initial marking $\mu_p(0) = \mu_{p_0}$, an **admissible constraint** satisfies two conditions:

1. $l^T \mu_{p_0} \leq b$, and

2. For all $\mu_p(N)$ reachable from $\mu_p(0)$ through any path of consecutively reachable markings, $\mu_p(0) \rightarrow \mu_p(1) \rightarrow \cdots \rightarrow \mu_p(N)$, where

$$l^T \mu_p(i) \leq b, \text{for} 1 \leq i \leq N,$$

$\mu_p(N)$ is an admissible marking.

If a constraint does not satisfy both of these conditions, then it is **inadmissible.**

Proposition 3. **General constraint admissibility.** A constraint on the marking and/or firing behavior of a Petri net is *admissible* iff

1. The initial conditions of the plant satisfy the constraint, and

2. There exists a maximally permissive controller (constructed under the assumption that all transitions are controllable) that enforces the constraint and does not inhibit any uncontrollable transitions that would otherwise be enabled.

Corollary 4. **Place-constraint admissibility.** The single vector constraint $l^T \mu_p \leq b$ is *admissible* iff the controller with incidence matrix $D_c = -l^T D_p$ and initial marking $\mu_{c0} = b - l^T \mu_{p0} \geq 0$ will never attempt to disable an uncontrollable transition that would otherwise be enabled.

Corollary 5. **Redundant place-constraints are admissible.** Given the single vector constraint

$$l^T \mu_p \leq b$$

on the marking of a plant with incidence matrix D_p and initial marking μ_{p0}, if there exits a place invariant x such that for all reachable μ_p, $x^T \mu_p = x^T \mu_{p0}$ implies that $l^T \mu_p \leq b$ is also true, then the constraint is admissible.

3 Direct Realization of Firing Vector Constraints

Assume that the plant must satisfy constraint (2). The direct interpretation of this constraint implies that transition t_j cannot fire if place p_i is marked, and, of course, place i can never contain more than one token. To bring this constraint to a form that contains elements of the marking vector only, the plant is transformed as follows. Transition j is replaced by two transitions and a place between them, as shown in Figure 1. This transformation is artificial and will not affect the Petri net model of the process. *Its sole purpose is to introduce the place p_j', which records the firing of the transition t_j.* After the controller has been computed the plant will be transformed back to its original form.

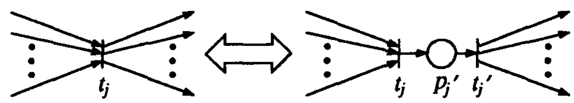

Figure 1: Transformation of a Transition.

The marking μ_j' of p_j' replaces q_j in constraint (2), which becomes

$$\mu_i + \mu_j' \leq 1 \tag{3}$$

The constraint now contains only μ's and a controller can now be computed. After the controller structure is computed, the two transitions and the place of the transformation collapse to the original transition thus restoring the original form of the plant while maintaining the enforcement of the new constraint. The same transformation is done to all the transitions that appear in the constraints. Constraints that contain only q's, i.e., constraints on allowable firing vectors with no concern for specific markings, are treated the same. Unlike standard invariant based controllers, *these controllers may contain self loops to the transitions indicated in the constraints.* Separate incidence matrices, D_c^+ and D_c^-, must be maintained for input and output halves of the controller.

Formally, given a plant (D_p, μ_{p0}) and constraint

$$l^T \mu_p + f^T q \leq b, f \geq 0 \tag{4}$$

the invariant based controller $(D_c = D_c^+ - D_c^-, \mu_{c0})$ is given by

$$D_c^+ = D_{lc}^+ + \max(0, D_{fc}^+ - D_{lc}^-) \tag{5}$$

$$D_c^- = \max(D_{lc}^-, D_{fc}^-) \tag{6}$$

$$\mu_{c0} = b - l^T \mu_{p_0} \tag{7}$$

where

$$D_{fc}^+ = D_{fc}^- = f^T \tag{8}$$

and

$$D_{lc}^+ = \max(0, D_{lc}) \tag{9}$$

$$D_{lc}^- = \max(0, -D_{lc}) \tag{10}$$

where

$$D_{lc} = -l^T D_p \tag{11}$$

and the notation $\max(a, b)$ refers to a vector constructed from the maximum elements in an element-by-element comparison of the two argument vectors.

The remainder of this subsection provides an analysis of the admissibility of firing vector constraints using the direct interpretation. Similar to Corollary 4, the following corollary defines when a constraint on the firing vector of a Petri net is admissible. This corollary assumes a more general definition of admissibility that indicates a constraint is admissible if its direct, optimal enforcement would not involve improper interaction with the uncontrollable/unobservable transitions of the net.

Corollary 6. **Transition-constraint admissibility.** The single vector constraint $f^T q \leq b$, where $f, b \geq 0$, is *admissible* under direct transition-constraint implementation on a plant with controllable transitions T_c, if $\forall j$ s.t. $f_j \neq 0, t_j \in T_c$.

Proof. The proof is by Proposition 3 on general constraint controllability. The direct transition-constraint enforcement method for the constraint $f^T q \leq b$ is maximally permissive since it is constructed as an invariant based controller. The initial marking of the controller $\mu_{c0} = b$ is valid if $b \geq 0$. The incidence matrix of the controller $D_c^+ = D_c^- = f^T$ contains input arcs to all transitions j such that $f_j \neq 0$. If all of these transitions are controllable, then the controller draws no arcs to uncontrollable transitions and the constraint is admissible. □

The admissibility of combined marking/firing constraints, $l^T \mu_p + f^T q \leq b$, will be discussed for the situation in which the constraints are *uncoupled.*

Definition 7. A constraint of the form (4), where $f \geq 0$, is called **uncoupled** if

$$T_l \cap T_f = \emptyset$$

where T_l is the set of transitions that are connected to the controller induced by the $l^T \mu_p$ portion of the constraint (transitions t_j such that $D_{lc}(j) \neq 0$ in equation (11)), and T_f is the set of transitions connected to the controller induced by the $f^T q$ portion of the constraint (transitions t_j such that $f_j \neq 0$).

Constraint (4) is uncoupled if the transitions involved in the $l^T \mu_p$ and $f^T q$ portions of the constraint are mutually exclusive.

Proposition 8. **Uncoupled place/transition constraints.** A vector constraint of form (4) is uncoupled iff

$$\forall i \text{ s.t. } f_i \neq 0, l^T D_p e_i = 0 \tag{12}$$

where e_i is a zero-vector with a 1 in the i^{th} place.

Proof. The set of plant transitions that will contain arcs to or from the controller is determined from the controller synthesis equations. This set is the union of the transitions connected by arcs induced by the $l^T \mu_p$ and $f^T q$ portions of the constraint, i.e., $T_f \cup T_l$. Equation (8) indicates that

$$T_f = \{t_j | f_j \neq 0\} \tag{13}$$

and equations (9) and (10) show

$$T_l = \{t_j | l^T D_p e_j \neq 0\} \tag{14}$$

Combining these with condition (12) implies

$$T_l \cap T_f = \emptyset \tag{15}$$

The sets of transitions used by the two portions of the controller are mutually exclusive and the constraint is uncoupled according to definition 7.

It is easy to see that if the constraints are uncoupled, i.e. $T_l \cap T_f = \emptyset$, then (12) must be true by working backward through the development above. If (12) were not true, then there would exist some $t_i \in T_f$ and $t_i \in T_l$, which would imply through equations (13) and (14) that $T_l \cap T_f \neq \emptyset$ and the constraints were coupled. □

Proposition 9. **Place/transition constraint admissibility.** An uncoupled vector constraint of form (4) is to be imposed on a plant (D_p, μ_{p0}) with uncontrollable transitions T_{uc} and controllable transitions $T_c, T_{uc} \cap T_c = \emptyset$.
if the constraints

$$l^T \mu_p \leq b \tag{16}$$
$$f^T q \leq |b| \tag{17}$$

are both admissible then $l^T \mu_p + f^T q \leq b$ is admissible.

Proof. If the admissibility of constraints (16) and (17) imply that (4) is admissible, then the inadmissibility of (4) will imply that either (16) or (17) is inadmissible or both. For $l^T \mu_p + f^T q \leq b$ to be inadmissible, it must lie outside the range of the plant's initial conditions, or a maximally permissive controller that enforces the constraint would attempt to inhibit an otherwise enabled transition in the set T_{uc}. Because (4) is uncoupled, the transitions that are connected to the controller places, T_f and T_l, are mutually exclusive. This means that at least one of the following three cases must be true for $l^T \mu_p + f^T q \leq b$ to be inadmissible.

1. The initial conditions of the plant violate the constraint.
2. The controller would attempt to inhibit a transition $t_j \in T_{uc}$, where $t_j \in T_f$.
3. Or the controller would attempt to inhibit an otherwise enabled transition $t_j \in T_{uc}$, where $t_j \in T_l$.

Case 1: The initial state of the plant is μ_{p0}. The firings indicated by the vector q are determined after the system commences its run, thus if the initial conditions of the plant violate constraint (4), then

$$l^T \mu_{p0} > b$$

This condition would also indicate that the constraint $l^T\mu_p \le b$ is inadmissible according to Corollary 4.

Case 2: According to the construction of the maximally permissive controller for direct transition constraints, the transitions in the set T_f are identical to the transitions that receive controller arcs in the constraint $f^Tq \le |b|$. If the controller attempts to inhibit an uncontrollable transition in this set, then the constraint $f^Tq \le |b|$ is inadmissible according to Corollary 6.

Case 3: The construction of the maximally permissive controller for the constraint $l^T\mu_p \le b$ shows that the transitions that receive controller arcs for this constraint are identical to the set T_l. If the controller for constraint (4) attempts to disable an otherwise enabled transition in the set T_l, then the constraint $l^T\mu_p \le b$ will be inadmissible according to Corollary 4.

Thus if both $l^T\mu_p \le b$ and $f^Tq \le |b|$ are admissible, then $l^T\mu_p + f^Tq \le b$ is also admissible. □

4 Indirect Realization of Firing Vector Constraints

Firing vector constraints can be realized by preventing the states that would allow the undesirable transition firing. This situation is analogous to the case when a transition is uncontrollable but is involved with regular marking constraints. Illegal states are prevented in the presence of uncontrollable transitions by preventing those states which could lead, through uncontrollable transitions, to the explicitly forbidden states. The results from [3–5] for handling uncontrollable transitions can be applied to constraints involving the firing vector through utilization of the graph transformations discussed in the previous section.

The procedure is illustrated in the example below.

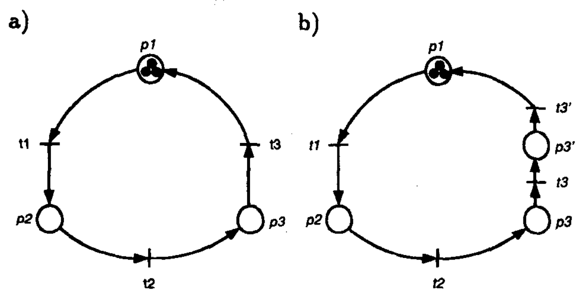

Figure 2: a) A simple net that will have a firing constraint enforced. b) The graph-transformed net of the net in Figure 2a.

Example. For the plant of Figure 2a, we wish to enforce the constraint

$$\mu_2 + q_3 \le 1 \tag{18}$$

Because the Petri net is so simple, we can see by inspection that the job can be done by enforcing the constraint $\mu_2 + \mu_3 \le 1$. But how can this new constraint be generated automatically based on (18)?

Suppose we perform the graph transformation on this net as shown in Figure 2b. The transformation changes (18) to

$$\mu_2 + \mu_3' \le 1 \tag{19}$$

If we continue to follow the procedure described in section 3, we would end up with a controller that directly enables and disables transition t_3. In order to prevent this from occurring, we will label transition t_3 as uncontrollable and then continue with the procedure.

Applying a constraint transformation method that accounts for uncontrollable transitions (such as those found in [3–5]) to (19), we obtain the following transformed constraint:

$$\mu_2 + \mu_3 + \mu_3' \le 1 \tag{20}$$

The controller that enforces this constraint can be automatically generated using the place invariant method and is shown in Figure 3a.

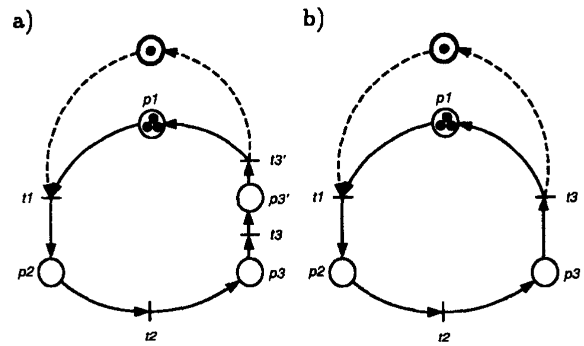

Figure 3: a) The net of Figure 2b with its Petri net controller. b) The untransformed net of Figure 2a with its Petri net controller.

The final stage is then to collapse the controlled net back to the form it had before the graph transformation was performed. The final controlled version of the net is shown in Figure 3b. Transition t_3 will not fire when place p_2 contains a token because the controller only allows one token at a time in places p_2 and p_3, which is the desired result.

The procedure used in the example is summarized below. Given a constraint

$$l^T\mu_p + f^Tq \le b \tag{21}$$

(where l may be zero, indicating a constraint on the firing vector alone,) first perform a transformation of the plant such that each transition specified by a nonzero entry in f includes a dummy place to mark its firing as described in section 3. The marking vector μ' is associated with the dummy places and the constraint becomes

$$\begin{bmatrix} l^T & f^T \end{bmatrix} \begin{bmatrix} \mu_p \\ \mu' \end{bmatrix} \le b \tag{22}$$

Next mark all transitions specified by nonzero entries in f as uncontrollable. Use established techniques [3–5] for the handling of uncontrollable transitions to find an admissible constraint that enforces the inadmissible constraint (22) and construct a supervising controller for this constraint. This will have the effect of preventing the states that could lead to (22) being violated. It will prevent the transitions specified by f from being enabled such that constraint (21) could be violated. Finally, collapse the net back to its original form by removing the dummy places and extra transitions as described in section 3.

5 Example: The Three Tanks Problem

Consider a plant of three fluid-filled tanks and a pump that can add fluid to each of them [2,5]. Tank i drains at a constant rate d_i. The pump can be moved to fill any of the three tanks. It fills tanks at a constant rate F. The control goal is to make sure that the fluid level in each of the three tanks stays within a safe boundary area. The fluid in tank i is not to rise above h_i nor is it to drain below l_i. Nothing is specifically stated about the dynamics of the flowing fluid. The inputs coming into the controller are events indicating fluid levels, and the control output is one of three discrete commands. For these reasons, the problem can be handled with DES supervisory control.

There are three states of interest with regard to any tank: 1) Tank fluid level is greater than pseudo overflow, 2) Tank fluid level is OK, 3) Tank fluid level is less than pseudo underflow. There are four relevant events associated with the evolution of any given tank: 1) Tank fluid level has raised past pseudo overflow, 2) Tank fluid level has lowered below pseudo overflow, 3) Tank fluid level has lowered below pseudo underflow, 4) Tank fluid level has raised above pseudo underflow. These events are uncontrollable, they occur upon observation of processes in the plant. There are three controllable events, u_1, u_2, and u_3, associated with the placement of the pump. These states and events are combined to form the Petri net model of Figure 4. The meaning of each place and transition within the model is defined in Table 1.

Using the base model of the system, we will synthesize a controller Petri net such that the control (u_1, u_2, or u_3) issued to the plant is equivalent to the firings of t_1, t_2, or t_3. The first step is to determine the constraints on the plant model. The first constraint is induced by the physical realities of the plant: only one pump can be serviced at a time.

$$q_1 + q_2 + q_3 \leq 1 \tag{23}$$

Transitions 1, 2, and 3 are all controllable, so the direct enforcement (section 3) of constraint (23) is admissible by Corollary 6 and may be implemented.

We need to insure that the tanks will not overflow, this means that we can not deliver fluid to tank i when tank

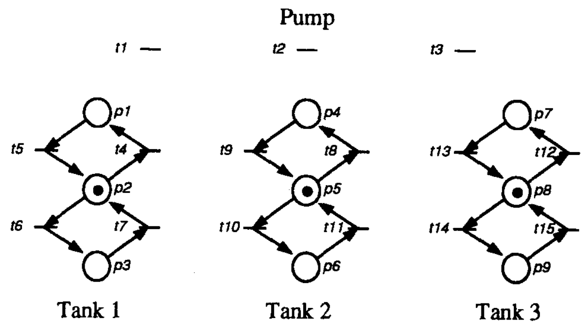

Figure 4: A Petri net model of the relevant events and states for the fluid-filled tank problem.

Controllable transitions	
t_1	Move pump to tank 1 and fill.
t_2	Move pump to tank 2 and fill.
t_3	Move pump to tank 3 and fill.
Uncontrollable transitions	
t_4, t_8, t_{12}	Pseudo overflow occurred.
t_5, t_9, t_{13}	Pseudo overflow cleared.
t_6, t_{10}, t_{14}	Pseudo underflow occurred.
t_7, t_{11}, t_{15}	Pseudo underflow cleared.
Places	
p_1, p_4, p_7	Pseudo overflow.
p_2, p_5, p_8	Fluid level OK.
p_3, p_6, p_9	Pseudo underflow.

Table 1: Explanation of transitions and places in Figure 4.

i is experiencing pseudo overflow.

$$q_1 + \mu_1 \leq 1 \tag{24}$$
$$q_2 + \mu_4 \leq 1 \tag{25}$$
$$q_3 + \mu_7 \leq 1 \tag{26}$$

Before implementing these constraints, we must first determine that they are admissible. This procedure starts by determining if each constraint is *uncoupled* (see Definition 7 and Proposition 8).

For constraint (24), the set T_f corresponds to the q_1 portion of the constraint and T_l corresponds to μ_1 portion of the constraint. Using the rules for controller construction and/or Proposition 8 we see that $T_f = \{t_1\}$ and $T_l = \{t_4, t_5\}$, thus $T_l \cap T_f = \emptyset$ and the constraint is uncoupled. The same can be shown for constraints (25) and (26). Proposition 9 indicates that if the two constraints $\mu_1 \leq 1$ and $q_1 \leq 1$ are admissible, then (24) is admissible. Corollary 6 shows that $q_1 \leq 1$ is admissible, since the transition it effects (t_1) is a controllable transition. Next the admissibility of $\mu_1 \leq 1$ must be verified.

A structural analysis of the plant indicates the presence of the following place invariant:

$$\mu_1 + \mu_2 + \mu_3 = 1 \tag{27}$$

Since equation (27) is always true throughout the evolution of the plant, $\mu_1 \leq 1$ is always true, which implies that $\mu_1 \leq 1$ is admissible according to Corollary 5. Having determined that $q_1 \leq 1$ and $\mu_1 \leq 1$ are both admissible, we have satisfied both conditions in Proposition 9 and have demonstrated that constraint (24) is indeed admissible. *The controller associated with this constraint will direct an arc to the uncontrollable transition t_4. However Proposition 9 and Corollary 5 insure that this arc will only be used for observation and never for inhibition.* A similar application of Proposition 9 indicates that constraints (25) and (26) are also admissible.

Finally we need to prevent underflow, which means that the pump must shift to the tank that has sent a "pseudo-underflow occurred" event. The analysis is omitted here, but the final constraints are

$$\begin{aligned} 2q_1 + \mu_6 + \mu_9 - 2\mu_3 &\leq 2 \quad (28)\\ 2q_2 + \mu_3 + \mu_9 - 2\mu_6 &\leq 2 \quad (29)\\ 2q_3 + \mu_3 + \mu_6 - 2\mu_9 &\leq 2 \quad (30)\end{aligned}$$

The admissibility of these constraints can be verified using Proposition 9. A controller for enforcing the constraints, (23),(24)–(26),(28)–(30), is automatically generated using the rules for direct implementation of firing vector constraints (section 3). The controlled plant is shown in Figure 5.

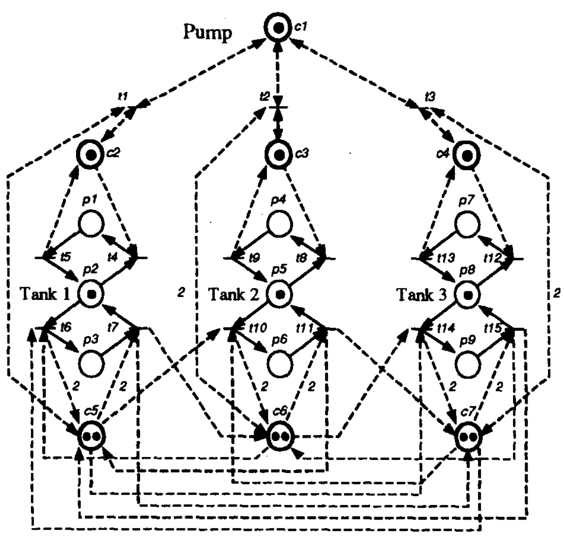

Figure 5: The plant of Figure 4 with added control structures.

6 Conclusions

A wide variety of supervisory control goals can be handled by representing them as linear predicates on the set of allowed Petri net states and enforcing these inequalities with invariant based controllers. Inequality (1) is not only appropriate for formulating a large range of forbidden state problems, generalized mutual exclusion constraints [1], and finite resource management problems, but it is also appropriate for specifying a number of supervision goals that are not normally thought of as state-based constraints. This paper extends the method to handle constraints written in terms of the firing vector as well as the marking, allowing constraints dealing with plant events to be specified directly. Temporary transformations of the graph structure of the plant were used to frame the event-based problem in terms of (1), allowing the methods of invariant based control to be employed. Two distinct approaches to enforcing these event-based constraints were presented here and were integrated with the concept of admissible constraints in order to handle uncontrollable and unobservable transitions within the plant. Which technique to use, direct or indirect, depends on the needs of the particular control problem and plant, for example, the three-tank process control plant used the direct definition, however the automated guided vehicle example in [7] uses the indirect definition.

References

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

[2] G. Labinaz, K. Rudie, S. L. Ricker, N. Sarkar, and M. M. Bayoumi, "A hybrid system investigation of fluid-filled tanks", *Submitted to the IEEE Transactions on Automatic Control Special Issue on Hybrid Systems*, 1997.

[3] Y. Li and W. M. Wonham, "Control of vector discrete event systems II – controller synthesis", *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 512–530, March 1994.

[4] J. O. Moody and P. J. Antsaklis, "Supervisory control of Petri nets with uncontrollable/unobservable transitions", In *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 4433–4438, Kobe, Japan, December 1996.

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

[6] T. Murata, "Petri nets: Properties, analysis, and applications", *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[7] K. Yamalidou, J. O. Moody, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants", *Automatica*, vol. 32, no. 1, pp. 15–28, January 1996.