Proceedings of the American Control Conference
Albuquerque, New Mexico June 1997
0-7803-3832-4/97/$10.00 © 1997 AACC

Characterization of Feasible Controls for Petri Nets with Unobservable Transitions

John O. Moody and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556 USA
jmoody@maddog.ee.nd.edu

Abstract

Supervisory control of discrete event systems modeled by Petri nets involves enforcing a set of constraints on the state, or marking, of the plant Petri net. Unobservable transitions within the plant may force the control designer to alter these original constraints to account for the inability of the supervisor to act when one of these transitions fires. A method for characterizing the constraints and the associated controllers which can be realized in the face of unobservable transitions is presented in this paper. While these are preliminary results, the characterization can be used by the designer to determine which linear constraints can be implemented without change, which constraints need to be transformed, and how those constraints should be transformed.

1. Introduction

The representation of discrete event systems by ordinary Petri nets [9, 11] allows for the use of many powerful algebraic tools for the realization of supervisory controllers [10] for these systems. In particular, it is possible to enforce a set of constraints on the plant state $\mu_p \in \mathbb{Z}^m, \mu_p \geq 0$ of the form

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

where $L \in \mathbb{Z}^{n_c \times m}, b \in \mathbb{Z}^{n_c}$ and $\mathbb{Z}$ is the set of integers. The inequality in (1) is read, like all of the vector and matrix inequalities in this paper, with respect to each element on the corresponding left and right hand sides of the inequality. If all of the transitions within the plant Petri net are controllable and observable, then it has been shown ([8, 12]) that (1) can be enforced by a Petri net controller which produces a place invariant (see [9, 11]) on the closed loop plant-controller system.

The incidence matrix of the closed loop system, D, and its marking, μ, are given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \tag{2}$$

where $D_p \in \mathbb{Z}^{m \times n}$ is the incidence matrix of the plant, $D_c \in \mathbb{Z}^{n_c \times n}$ is the incidence matrix of the controller, and $\mu_c \in \mathbb{Z}^{n_c}$ is its marking. The controller and its initial marking μ_{c_0} is calculated using

$$D_c = -LD_p \tag{3}$$

$$\mu_{c_0} = b - L\mu_{p_0} \tag{4}$$

where $\mu_{p_0} \in \mathbb{Z}^m$ is the initial marking of the plant. Controllers constructed in this way are identical to the monitors introduced by Giua *et al.* [2]

Some sets of constraints can not be enforced and thus appropriate controllers do not exist. It is possible to enforce the set of constraints (1) iff

$$b - L\mu_{p_0} \geq 0 \tag{5}$$

The discussion above assumes that all of the transitions in the Petri net plant will permit observation and can be inhibited if the controller deems it necessary. Li and Wonham [3] have made important contributions involving the optimal (maximally permissive) transformation of an original set of marking constraints into a set which accounts for possible uncontrollable actions (transitions) within the plant net. An alternative method for generating transformations of constraints to account for uncontrollable transitions was introduced in [6]. This method is computationally more efficient than that presented in [3], however it always yields a linear transformation, and thus is not always optimal.

The concept of unobservable transitions is introduced into this framework in [5]. Unobservable transitions are defined on the Petri net graph because they represent the occurrence of a real event, but these events are either impossible or too expensive to detect directly. It is also possible, in the event of a sensor failure, that a transition might suddenly become unobservable, forcing a redesign or adaptation of the control law. It is illegal for the controller to change its state based on the firing of an unobservable transition, because there is no direct way for the controller to be told that such a transition has fired. Both input and output arcs from the controller places are used to change the controller state based on the firings of plant transitions, thus, when the entire closed loop system is to be represented by a Petri net, it is illegal for the places which represent the controller to draw arcs to any

*This research was partially funded by the National Science Foundation. Grant ECS95-31485.

of the unobservable transitions. This means that a Petri net controller can not inhibit the firing of an unobservable transition. In practice this is a reasonable restriction: it is uncommon in closed loop control to attempt to use an actuator for which the controller can receive no feedback. Thus, in this paper, unobservable transitions are assumed to be uncontrollable as well.

Section 2. of this paper presents a procedure for characterizing all of the linear constraints that may be realized on a Petri net with unobservable transitions. The characterization allows for the designer to determine, often times by inspection, which linear constraints can be realized, which need to be transformed, and how those constraints need to be transformed. An example is presented in section 3., and concluding remarks appear in section 4..

2. Characterization of Feasible Transformations and Controls

It is illegal for a controller to change its state based on the firing of an unobservable transition in the plant. This restriction means that the realization of certain control goals must be adapted to account for unobservability. One method for realizing this adaptation is to transform the original constraints on the plant such that the new constraints would not cause a controller to use these transitions. Computational techniques for adapting plant constraints in the face of unobservable transitions have been described in [5]. Section 2.1. provides a method for characterizing all linear constraints which can be legally enforced on a system with unobservable transitions, and section 2.2. shows how this characterization can be used to create legal controllers for realizing particular desired constraints.

2.1. Characterization of Linear Constraint Transformations

Equation (3) shows that it is possible to construct the incidence matrix D_c of a maximally permissive Petri net controller as a linear combination of the rows of the incidence matrix of the plant. Non zero elements in the columns of D_c correspond to transition firings for which the controller will change its own internal state and, for negative elements in D_c, possibly introduce inhibitions on the plant's behavior. Both input and output arcs from the controller places are used to change the controller state based on the firings of plant transitions. Let the matrix D_{uo} represent the incidence matrix of the unobservable portion of the Petri net. This matrix is composed of the columns of D_p which correspond to unobservable transitions. It is illegal for the controller $D_c = -LD_p$ to contain any arcs in the unobservable portion of the net, thus an enforceable set of constraints will satisfy

$$LD_{\text{uo}} = 0 \tag{6}$$

Any L which satisfies (6) will lie within the kernel of D_{uo}. Let X satisfy

$$XD_{\text{uo}} = 0 \tag{7}$$

where X is an integer matrix with dimension $(m - \text{rank } D_{\text{uo}}) \times m$. The rows of X form a linearly independent basis for the kernel of D_{uo} (X is full rank). The process of finding X is equivalent to finding the minimal support place invariants (an algorithm appears in [4]) of the unobservable portion of the plant Petri net. All realizable constraints must lie within the basis described by the rows of X, and thus can be formed as linear combinations of these rows. Therefore every feasible constraint can be described by $k^T X$ where k is an integer vector with dimension $(m - \text{rank } D_{\text{uo}})$. In general, the coefficient matrix of any set of feasible constraints $L' \in \mathbb{Z}^{n_c \times m}$ can be written

$$L' = KX \tag{8}$$

where $K \in \mathbb{Z}^{n_c \times (m - \text{rank } D_{\text{uo}})}$. Equation (3) can then be used to calculate the incidence matrix of the controllers which will enforce these constraints:

$$D_c = -KXD_p \tag{9}$$

2.2. Realizing Constraint Transformations

Suppose we have a set of constraints $L\mu \leq b$ such that $LD_{\text{uo}} \neq 0$. It is necessary to create new constraint matrices (L', b') with two properties.

1. $L'D_{\text{uo}} = 0$

2. $L'\mu_p \leq b' \rightarrow L\mu \leq b$

Property 1 is necessary to insure that the new controller will not utilize the unobservable transitions. Section 2.1. shows how to characterize all such matrices L'. Property 2 indicates that the new constraints must be at least as restrictive as the original ones. We can't have the new constraints allowing states that we originally wanted to prohibit. In order to deal with this condition, the following lemma from [6] is used.

Lemma 1.

$$\text{Let } R_1 \in \mathbb{Z}^{n_c \times m} \text{ satisfy } R_1\mu_p \geq 0 \;\forall\; \mu_p. \tag{10}$$

$$\text{Let } R_2 \in \mathbb{Z}^{n_c \times n_c} \text{ p.d.d.} \tag{11}$$

Where p.d.d. means R_2 is a positive definite diagonal matrix. If $L'\mu_p \leq b'$ where

$$L' = R_1 + R_2 L \tag{12}$$

$$b' = R_2(b + 1) - 1 \tag{13}$$

and 1 is an n_c dimensional vector of 1's, then $L\mu_p \leq b$.

Thus to do the transformation, it is necessary to determine values for the matrices R_1 and R_2 which meet assumptions (10) and (11). Computational techniques for determining these matrices are given in [5], however it is possible for a designer to determine the values of R_1 and R_2 by using the kernel of D_{uo}. Combining equations (8) and (12) we see that

$$L' = KX = R_1 + R_2 L$$

The designer should premultiply each constraint in L by some positive integer (which will determine the diagonal elements in R_2) and add new positive coefficients (which will determine R_1) such that the new constraint is a linear combination of the rows of X. This process will yield the L' matrix, and b' can be calculated using R_2 and equation (13).

3. Example: Piston Rod Robotic Assembly Cell

This example of a piston rod assembly cell is borrowed from chapter 8 of [1] and the derivation of the controller used is given in [7]. The Petri net model of the plant and its controller is shown in Fig. 1. Table 1 details the meaning of each place in the net. A token in any of the Petri net places signifies that the action or condition specified in Table 1 is taking place. The piston rod assembly is performed by two robots, and the primary feedback mechanism is a vision system. An S-380 robot is used to prepare and align the parts for assembly, and an M-1 robot installs the cap on the piston rod. The specific duties of each robot are described below.

S-380: The S-380 robot remains idle until a new engine block and crank shaft become available. This event is represented by the appearance of a token in place p_1 in Fig. 1. The firing of transition t_1 indicates the start of the process. At this time the S-380 moves the crank shaft into alignment and brings a new piston rod into the work area. These actions are represented by places p_2 and p_3. The firing of transition t_3 indicates that the S-380 has completed its duties for the particular engine block.

M-1: The M-1 robot starts its duties by picking up a piston pulling tool (place p_4) and, assuming the S-380 has brought a piston rod into position, pulls the piston rod into the engine block and replaces the pulling tool (place p_5). The M-1 then picks up a cap and secures it to the piston rod using two bolts (places p_6 and p_7). The firing of transition t_8 indicates that the M-1 has successfully installed the cap and the engine block has been conveyed out of the work space. At this time work can begin on a new engine block.

	Plant Places
p_1	Work area clear, engine block, crank shaft ready.
p_2	S-380 robot aligns the crank shaft.
p_3	S-380 picks up piston rod and positions it.
p_4	M-1 robot picks up the piston pulling tool.
p_5	M-1 positions piston rod and returns pulling tool.
p_6	M-1 picks up a cap and positions it on piston rod.
p_7	M-1 fasten cap to piston rod using two bolts.
	Control Places
c_1	S-380 robot is available for work.
c_2	M-1 robot is available for work.
c_3	S-380 robot has completed preparations.
c_4	A piston rod is available.
c_5	The piston pulling tool is available.
c_6	A cap is available.
c_7	Two nuts are available.

Table 1: Place descriptions for the piston rod assembly Petri net of Fig. 1.

The incidence matrix, D_p, and initial marking, μ_{p0}, of the plant are given by

$$D_p = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$
$$\mu_{p0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{14}$$

It is necessary to insure that the robots' activities are synchronized and that the finite resources, including the robots themselves, are properly accounted for. The following constraints are placed on the plant.

$$\begin{aligned} \mu_2 + \mu_3 &\leq 1 \\ \mu_4 + \mu_5 + \mu_6 + \mu_7 &\leq 1 \\ \mu_1 + \mu_2 + \mu_3 + \mu_5 + \mu_6 + \mu_7 &\leq 1 \\ \mu_3 &\leq 1 \\ \mu_4 + \mu_5 &\leq 1 \\ \mu_5 + \mu_6 &\leq 1 \\ \mu_5 + \mu_6 + \mu_7 &\leq 1 \end{aligned} \tag{15}$$

The derivation of these constraints can be found in [7]. Each of these constraints yields an individual place in the controller. A description of the duties of each of the control places is given in table 1. The controller was generated by an application of the technique described in section 1.. The incidence matrix and initial marking of the controller

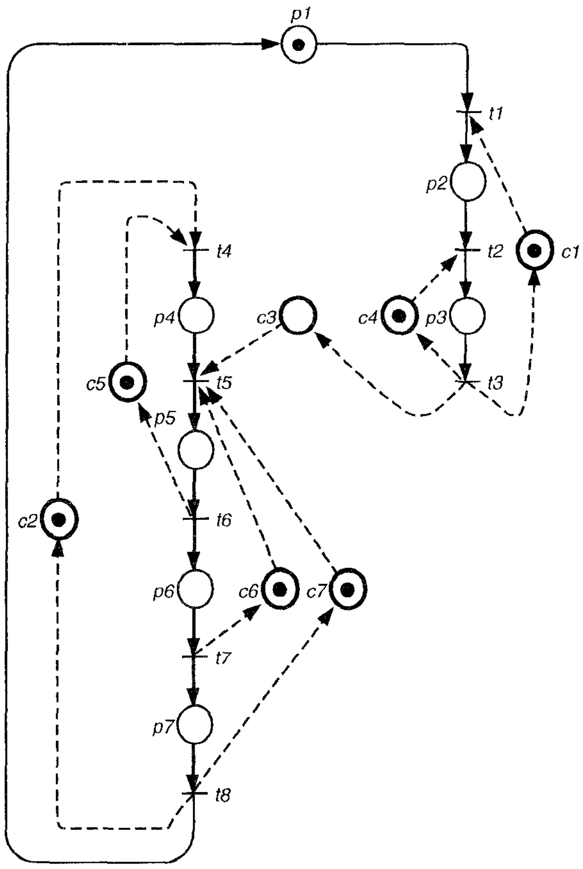

Figure 1: The assembly cell model with Petri net controller.

are

$$Dc = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$
$$\mu_{c0} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T \tag{16}$$

The piston rod assembly cell presented in [1] uses a vision system to provide sensory feedback to the controller. Suppose that an obstruction has appeared between the camera and the work space, partially obscuring the view of the M-1 robot's area. The controller can still observe the M-1 robot starting and completing its task, but it can no longer track the robot while it performs its duties. Transitions t_5, t_6, and t_7 have become unobservable. This means that there should be no arcs from any of these transitions to the controller places, however it can be seen from (16) that the current version of the controller incidence matrix contains nonzero elements in columns five through seven. Let D_{uo} be a matrix composed of the unobservable columns of D_p, in this example D_{uo} is composed of the fifth, sixth, and seventh columns of D_p. In order to compensate for the sensor failure, we will first find the kernel of D_{uo}. There are seven rows in D_{uo}, and the rank of the matrix is three. This indicates that the kernel X will have $7 - 4 = 3$ rows.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}}_{D_{\text{uo}}} = 0$$

Rows one through three of X tell us that constraints involving only μ_1, μ_2, and μ_3 are independent and will not have to be transformed in order to meet the unobservability requirements. However, row four of X indicates that the coefficients on μ_4, μ_5, μ_6, and μ_7 must be equal. It is now a simple matter to rewrite the set of constraints (15) in order to meet this requirement:

$$\begin{aligned} &\mu_2 + \mu_3 \le 1 \Rightarrow \text{Unchanged} \\ &\mu_4 + \mu_5 + \mu_6 + \mu_7 \le 1 \Rightarrow \text{Unchanged} \\ &\mu_1 + \mu_2 + \mu_3 + \mu_5 + \mu_6 + \mu_7 \le 1 \Rightarrow \\ &\qquad \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 \le 1 \\ &\mu_3 \le 1 \Rightarrow \text{Unchanged} \\ &\mu_4 + \mu_5 \le 1 \Rightarrow \mu_4 + \mu_5 + \mu_6 + \mu_7 \le 1 \\ &\mu_5 + \mu_6 \le 1 \Rightarrow \mu_4 + \mu_5 + \mu_6 + \mu_7 \le 1 \\ &\mu_5 + \mu_6 + \mu_7 \le 1 \Rightarrow \mu_4 + \mu_5 + \mu_6 + \mu_7 \le 1 \end{aligned} \tag{17}$$

In all cases, each transformed constraint can be realized by simple additions of new coefficients to the original constraints in (15). This insures that the conditions of lemma 1 are obeyed: the new constraints will not allow states prohibited by the originals.

The constraints in (17) are now used to generate a reconfigured controller. The incidence matrix and initial marking are

$$Dc = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$
$$\mu_{c0} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T$$

Observe how the fifth through seventh columns of D_c have been completely zeroed. The reconfigured control is shown in figure 2.

Note that the last three transformed constraints in (17) are identical, however all constraints will be implemented separately in the reconfigured version of the controller. This is because the different controller places have different interpretations (see table 1) and because we intend to transform the controller arcs back to their original configurations once the sensor obstruction has been removed.

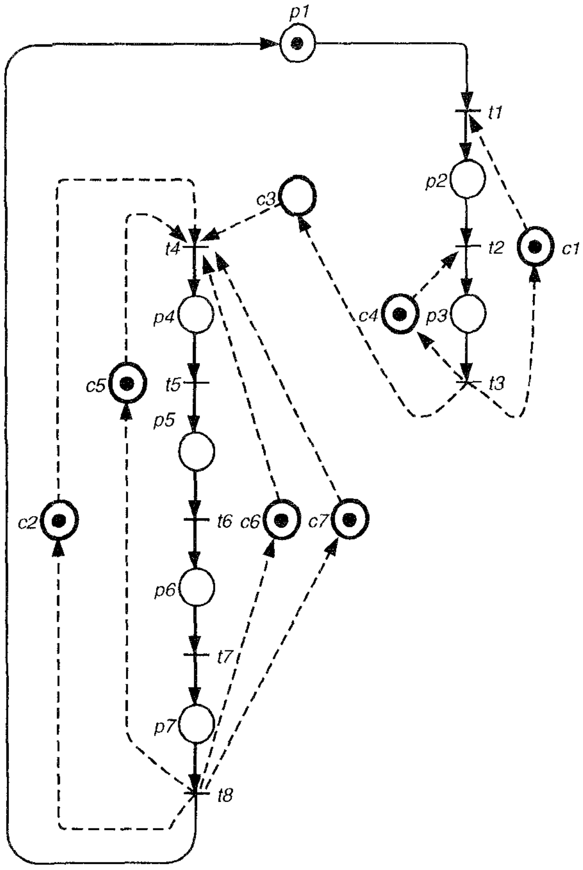

Figure 2: The assembly cell model with a controller that accounts for a sensor loss making transitions t_5, t_6 and t_7 unobservable.

4. Conclusions

This paper is part of an ongoing project in the formulation and formalization of a discrete event system control design procedure that relies on simple and efficient linear algebraic techniques. A method for characterizing the set of all realizable linear constraints in the face of unobservable transitions within the plant has been presented. The characterization of feasible constraints includes enough information so that it is often possible for the control designer to determine appropriate transformations of existing constraints simply by inspection.

References

[1] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri Nets in Manufacturing Systems*, IEEE Press, Piscataway, New Jersey, 1995.

[2] 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.

[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. Martinez and M. Silva, "A simple and fast algorithm to obtain all invariants of a generalized Petri net", In *Advances in Petri Nets*, number 52 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1980.

[5] 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.

[6] J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Feedback Petri net control design in the presence of uncontrollable transitions", In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 1, pp. 905–906, New Orleans, LA, December 1995.

[7] J. O. Moody, P. J. Antsaklis, and M. D. Lemmon, "Petri net feedback controller design for a manufacturing system", In *Proceedings of the IFAC 13th Triennial World Congress*, volume B, pp. 67–72, San Francisco, CA, July 1996.

[8] J. O. Moody, K. Yamalidou, M. D. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants", In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 3, pp. 3104–3109, Lake Buena Vista, FL, December 1994.

[9] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Engelwood Cliffs, NJ, 1981.

[10] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems", *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–97, 1989.

[11] W. Reisig, *Petri Nets*, Springer-Verlag, Berlin; New York, 1985.

[12] 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.