Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

Supervisory Control of Petri Nets with Uncontrollable/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

This paper expands upon results of previous research dealing with the supervisory control of Petri net modeled discrete event systems that contain uncontrollable transitions. The concept of unobservable plant transitions is introduced here and incorporated into the controller design procedure. New conditions are developed which govern the existence of controllers for these problems. Two procedures are presented for automatically generating controllers for plants that incorporate uncontrollable and unobservable events.

1 Introduction

The representation of discrete event system by ordinary Petri nets [1,2] allows for the use of many powerful algebraic tools for the realization of supervisory controllers [3] 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 ([4,5]) that (1) can be enforced by a Petri net controller which produces a place invariant (see [1,2]) on the closed loop plant-controller system. The Petri net incidence matrix, $D_c \in Z^{n_c \times n}$, of the controller is given by

$$D_c = -LD_p \tag{2}$$

where $D_p \in \mathbb{Z}^{m \times n}$ is the incidence matrix of the plant. The initial marking of the controller, $\mu_{c_0} \in \mathbb{Z}^{n_c}$ is

$$\mu_{c_0} = b - L\mu_{p_0} \tag{3}$$

where $\mu_{p_0} \in \mathbb{Z}^m$ is the initial marking of the plant. The incidence matrix, D, and marking, μ, of the closed loop plant-controller system are given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \tag{4}$$

Controllers constructed in this way are identical to the monitors introduced by Giua *et al.* [6]

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 [7,8] 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 within the plant net. The uncontrollable actions correspond to transitions within the plant Petri net that the controller has no power to inhibit. In general, the fact that a constraint is linear does not imply that the maximally permissive version of this constraint that accounts for uncontrollable transitions will also be linear. However Li and Wonham have presented sufficient conditions dealing with the structure of the uncontrollable portion of the plant that indicate when this situation will occur.

An alternative method for generating transformations of constraints to account for uncontrollable transitions was introduced in [9]. This method is computationally more efficient than that presented in [8], however it always yields a linear transformation, and thus is not always optimal. Section 2 of this paper expands upon these results by including the idea of unobservable transitions. Unobservable transitions provide no information to the controller when they fire. When they are present in the plant, it is necessary to transform constraints in a similar manner as that required for uncontrollable transitions. These results as well as new conditions for the existence of valid controllers and

0-7803-3590-2/96 $5.00

the necessity of certain transformation mechanics are presented. Section 3 proposes two procedures for automatically generating the transformations described in section 2. Concluding remarks appear in section 4.

2 Linear Constraint Transformations Due to Uncontrollable and Unobservable Transitions

Equation (2) in section 1 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. Negative elements in D_c correspond to arcs from controller places to plant transitions. These arcs act to inhibit plant transitions when the corresponding controller places are empty, and thus they can only be applied to plant transitions which permit such external control. If, as in the previous section, we group all of the columns of D_p which correspond to transitions which can not be controlled into the matrix D_{uc}, then, in order for a set of constraints to be consistent with the uncontrollable transitions in the plant, it must be true that the matrix LD_{uc} contains no positive elements, as these will correspond to controlling arcs when constructing the controller as $D_c = -LD_p$. An enforceable set of constraints will satisfy

$$LD_{\text{uc}} \leq 0 \tag{6}$$

It is also possible that transitions within the plant may be unobservable, i.e., they 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. The problem of handling unobservable transitions has been touched on in [10]. Here the constraints placed on a controller due to unobservable transitions are more rigorously defined, and a systematic method of dealing with them is proposed. It is illegal for the controller to change its state based upon 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. 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, just as D_{uc} is composed of the uncontrollable columns of D_p. 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

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

Conditions (6) and (7) indicate that it is possible to observe a transition that we can not inhibit, but it is illegal to directly inhibit a transition that we can not observe.

Suppose, given a set of constraints $L\mu_p \leq b$, we construct the matrices LD_{uc} and LD_{uo} and observe that there are violations to conditions (6) and/or (7). Since the controller is made of a linear combination of the rows of D_p, it is interesting to consider the situation where we use the addition of further rows from D_{uc} in order to eliminate the positive elements of LD_{uc} and use rows from D_{uo} to eliminate the nonzero elements of LD_{uo}, i.e., if we are going to use a place invariant forming Petri net controller, what additions to the constraints would we need to make in order to eliminate positive elements from LD_{uc} and nonzero elements from LD_{uo}? What constraints, of the form $L'\mu_p \leq b'$, that can be enforced by an invariant-based controller, will also maintain the original constraint $L\mu_p \leq b$ while not interfering with the uncontrollable/unobservable portions of the plant? The following lemma appeared in [9].

Lemma 1.

$$\text{Let } R_1 \in \mathbb{Z}^{n_c \times m} \text{ satisfy } R_1\mu_p \geq 0 \; \forall \; \mu_p. \tag{8}$$
$$\text{Let } R_2 \in \mathbb{Z}^{n_c \times n_c} \text{ p.d. diagonal matrix} \tag{9}$$

If $L'\mu_p \leq b'$ where

$$L' = R_1 + R_2 L \tag{10}$$
$$b' = R_2(b + \mathbf{1}) - \mathbf{1} \tag{11}$$

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

Lemma 1 shows a class of constraints, $L'\mu_p \leq b'$, which, if enforced, will imply that $L\mu_p \leq b$ are also enforced. The following lemma is used to show the conditions under which a particular set of constraints can be enforced on a particular Petri net.

Lemma 2. The constraint(s) $L'\mu_p \leq b'$, where $L' \neq 0$ and b' are defined by (10) and (11), can be enforced on a Petri net with initial marking μ_{p_0} iff

$$0 \leq R_1\mu_{p_0} \leq R_2(b + \mathbf{1} - L\mu_{p_0}) - \mathbf{1} \tag{12}$$

Proof. Substituting L' and b' into (12) gives $0 \leq b' - L'\mu_{p_0}$ which is equivalent to the condition $L'\mu_{p_0} \leq b'$, which states that the initial conditions of the plant must fall within the acceptable region of the constraints. Clearly, if a controller does exist, then the initial conditions of the plant must not violate the constraints. Furthermore, as shown in in [4–6], if the initial conditions lie within the acceptable region of the plant (inequality (5)), a controller to enforce the conditions can be computed with incidence matrix $D_c = -L'D_p$ and initial marking $\mu_{c_0} = b' - L'\mu_{p_0}$. □

Proposition 3 combines conditions (6) and (7) with the conditions for creating a valid set of transformed constraints in lemmas 1 and 2 to show how to construct a Petri net controller to enforce the constraints $L\mu_p \leq b$ which does not involve inappropriate interference with the uncontrollable and unobservable portions of the plant net.

Proposition 3. Let a plant Petri net with incidence matrix D_p be given with a set of uncontrollable transitions described by D_{uc} and a set of unobservable transitions described by D_{uo}. A set of linear constraints on the net marking, $L\mu_p \leq b$, are to be imposed. Assume R_1 and R_2 meet (8) and (9) with $R_1 + R_2 L \neq 0$ and let

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_{uc} & D_{uo} & -D_{uo} & \mu_{p0} \\ LD_{uc} & LD_{uo} & -LD_{uo} & L\mu_{p_0} - b - \mathbf{1} \end{bmatrix} \leq \begin{bmatrix} 0 & 0 & 0 & -\mathbf{1} \end{bmatrix} \tag{13}$$

Then the controller

$$D_c = -(R_1 + R_2 L)D_p = -L'D_p \tag{14}$$

$$\mu_{c_0} = R_2(b+\mathbf{1}) - \mathbf{1} - (R_1 + R_2 L)\mu_{p_0} = b' - L'\mu_{p_0} \tag{15}$$

exists and causes all subsequent markings of the closed loop system (4) to satisfy the constraint $L\mu_p \leq b$ without attempting to inhibit uncontrollable transitions and without detecting unobservable transitions.

Proof. According to (2) and (3), equations (14) and (15) define a controller that enforces the constraint $L'\mu_p \leq b'$. Lemma 1 shows that if assumptions (8) and (9) are met then a controller which enforces a particular constraint $L'\mu_p \leq b'$ will also enforce the constraint $L\mu_p \leq b$. The fourth column of inequality (13) indicates that the condition in lemma 2 is satisfied, thus the controller exists and the control law can be enforced. The first column of (13) indicates that $L'D_{uc} \leq 0$, thus condition (6) is satisfied and no controller arcs are drawn to the uncontrollable transitions. The second and third columns of (13) indicate that $L'D_{uo} = 0$, thus condition (7) is satisfied and no arcs are drawn between the controller places and the unobservable plant transitions. □

Note that $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$, which is used to describe the constraint transformation, multiplies from the left in (13), thus these matrices represent the use of rows from D_{uc} to eliminate positive elements from LD_{uc}, and the use of rows from D_{uo} to zero the elements of LD_{uo}, as discussed above.

3 Generating Constraint Transformations

The usefulness of proposition 3 for specifying controllers to handle plants with uncontrollable and unobservable transitions lies in the ease in which the matrices R_1 and R_2, with the appropriate properties, can be generated. Section 3.1 shows how the information in the proposition can be converted into an integer linear programming for determining R_1 and R_2, and section 3.2 proposes a scheme for determining valid R_1 and R_2 values by performing matrix row operations.

3.1 An Integer Linear Program

It is possible to convert the conditions in proposition 3 into an integer linear programming problem (ILP) in the standard form of

$$\min_x z(x) = x^T c$$
$$\text{s.t.} \begin{cases} x^T A &= d \\ x &\geq 0 \text{ (integer)} \end{cases} \tag{16}$$

We will consider only a single constraint on the system; multiple constraints can be handled individually and independently. Thus $n_c = 1$, L and R_1 are vectors, and b and R_2 are scalars.

In order to satisfy condition (8), we can specify that $R_1 \geq 0$, since we know that $\mu_p \geq 0$. In fact, it is necessary to specify that all of the elements of R_1 are greater than or equal to zero if the markings of the plant places are unbounded or they are bounded but the bound is not known.

Condition (9) states that we want $R_2 > 0$. In order to obtain variables that fit the conditions for x in (16), define

$$R_2' = R_2 - 1 \tag{17}$$

Since R_2 is an integer, $R_2' \geq 0$ implies that $R_2 > 0$.

Substituting the new variable R_2' into condition (6) yields

$$R_1 D_{uc} + R_2' L D_{uc} \leq -LD_{uc}$$

A vector of slack variables, $R_3 \geq 0$, is introduced in order to convert the inequality into an equality.

$$R_1 D_{uc} + R_2' L D_{uc} + R_3 = -LD_{uc}$$

R_3 is a column vector like R_1 but with dimension equal to the number of columns of D_{uc} (the number of uncontrollable transitions).

Substituting R_2' into condition (7) gives

$$R_1 D_{uo} + R_2' L D_{uo} = -LD_{uo}$$

which is already in the form of an equality, so an additional slack variable is unnecessary.

The new variable R_2' is now substituted into the condition given in lemma 2 which indicates whether the given constraint transformation can be implemented.

$$R_1\mu_{p0} + R_2'(L\mu_{p0} - (b+1)) \le b - L\mu_{p0} \tag{18}$$

Let

$$R = \begin{bmatrix} R_1 & R_2' & R_3 \end{bmatrix}$$

and the ILP can now be defined as

$$\min_R z(R) = R \begin{bmatrix} \mu_{p0} \\ L\mu_{p0} - b - 1 \\ 0 \end{bmatrix}$$

$$\text{s.t.} \begin{cases} R \begin{bmatrix} D_{uc} & D_{uo} \\ LD_{uc} & LD_{uo} \\ I & 0 \end{bmatrix} & = -L \begin{bmatrix} D_{uc} & D_{uo} \end{bmatrix} \\ R & \ge 0 \text{ (integer)} \end{cases} \tag{19}$$

which is in the form of (16).

After solving (19), if the minimum of the objective function $z^* = z(R^*)$ is greater than $b - L\mu_{p0}$ then the problem can not be solved as there are no values of R_1 and R_2 which will satisfy the condition in lemma 2. If the minimum is less than or equal to $b - L\mu_{p0}$, then transform R_2' back into R_2 and generate the controller using the formulae in proposition 3.

It is possible that there may be problems associated with this method of generating R_1 and R_2. For a controller to exist, we need the objective function of the ILP, $z(R) \le b - L\mu_{p0}$, however it not clear why we should attempt to minimize this function or what the results of such an attempt might be. In practical problems, the objective function may well be unbounded. In this case it is necessary for the designer to place an extra constraint on the problem to bound the objective function and obtain an answer. It is also possible that the ILP may yield the pathological transformation $L' = R_1 + R_2 L = 0$, when there are other nonzero possibilities for L'.

3.2 A Procedure Using Matrix Row Operations

It is possible to obtain appropriate constraint transformations by performing row operations on a matrix containing the uncontrollable and unobservable columns of the plant incidence matrix. The computational part of this procedure involves little more than the integer triangularization of a matrix, and thus it is simpler to compute R_1 and R_2 using this method than by using the ILP presented in the previous section. Before presenting the algorithm itself, the following terms are clarified:

D_{uc} : An $m \times n_{uc}$ matrix consisting of the columns of the plant incidence matrix D_p that correspond to transitions that are uncontrollable, but which may be observed. n_{uc} is the number of these transitions.

D_{uo} : An $m \times n_{uo}$ matrix consisting of the columns of D_p which are unobservable (just as defined in previous sections).

In the discussions in previous sections, D_{uc} may have included columns which were unobservable as well as uncontrollable, but here all of the columns of D_{uc} are observable. Conditions (6) and (7) show that unobservability implies stricter demands than uncontrollability, and in fact, any transition labeled as unobservable is also uncontrollable. D_{uc} is defined here as being strictly observable so that we can relax our requirements when dealing with this section of the matrix. Algorithm 1 presents the procedure for determining R_1 and R_2.

Algorithm 1 (Constraint Transformation).

```
Input:  L ∈ Z^{n_c×m}, b ∈ Z^{n_c}, D_uc ∈ Z^{m×n_uc},
  D_uo ∈ Z^{m×n_uo}, μ_p0 ∈ Z^m
if (LD_uc ≤ 0 and LD_uo = 0) then
   R_1 := 0_{n_c×m}, R_2 := I_{n_c×n_c}
else
   M := [ D_uc    D_uo    I ]
        [ LD_uc   LD_uo     ]
   Let M(i,j) be the (i,j)^th element of M.
   Zero all positive elements in the LD_uc
     portion of M using Algorithm 2.
   if M(m + 1...m + nc, 1...n_uc) has any
     positive elements then
      FAIL
   end if
   Zero the LD_uo portion of the M matrix
     using Algorithm 3.
   if M(m + 1...m + nc, n_uc + 1...n_uc + n_uo)
     has any nonzero elements then
      FAIL
   end if
   R_1 := M(m+1...m+n_c, n_uc+n_uo+1...n_uc+
     n_uo + m)
   R_2 := M(m + 1...m + n_c, n_uc + n_uo + m +
     1...n_uc + n_uo + m + n_c)
end if
L' := R_1 + R_2 L
b' := R_2(b + 1) − 1
if L'μ_p0 > b' then
   FAIL
end if
Output:  L' and b'.
```

As was done in section 3.1, we shall insure that condition (8) is met by making $R_1 \ge 0$. In terms of row operations, this means that elements in rows are eliminated strictly through addition, never through subtraction, and that rows can be premultiplied only by positive integers. The procedure for zeroing out the elements in a column of numbers which have the opposite sign of the "pivot" is given in Algorithm 4.

Algorithm 2 (Zeroing of positive elements in D_{uc}).

```
Input:  M ∈ Z^{(m+n_c)×(n_uc+n_uo+m+n_c)}
i := 1
while i ≤ min(n_uc, m)
   if any M(i...m, i) < 0 then
      Find row j in M(i...m, i) with a
        negative element.
      Exchange rows i and j in M
      Use Algorithm 4 to eliminate
        positive integers in
        M(i...m + n_c, i)
   else if any M(m+1...m+n_c, i) > 0 then
     FAIL
   end if
   i := i + 1
end while
Output M and i
```

Algorithm 1 insures that condition (9) is met because the procedure for choosing the "pivot" elements never picks from the LD_{uc} and LD_{uo} portions of the M matrix. Combined with the zeroing procedure of Algorithm 4, these steps insure that the R_2 portion of the M matrix is diagonal with strictly positive elements.

Algorithms 2 and 3 (called by Algorithm 1) are used to make sure that the transformed constraints meet conditions (6) and (7). The feasibility check at the end of Algorithm 1 directly tests the condition of lemma 2 to insure that the controller does exist. The instructions for picking positive or negative elements to act as pivots in the two main loops are left specifically vague. Different methods of choosing the pivot will lead to different constraint transformations. It would be possible, for instance, to find a basis for all valid constraint transformations by repeating the procedures in Algorithm 1 whenever there was a choice of more than one pivot for a given column.

Even if all the possible pivots are used in Algorithm 2, there may still be other R_1 and R_2 values which yield transformed constraints that meet condition (6). Algorithm 2 forces all positive elements in the LD_{uc} portion of the matrix to go to zero. However condition (6) states that we need $LD_{uc} \leq 0$, so the question is, is it ever desirable or necessary to transform positive elements of LD_{uc} into negative elements? Should Algorithm 2 incorporate this ability? At this time, this question can not be answered definitely, however the following points, starting with the lemma below, are presented to shed light on the question.

Algorithm 3 (Zeroing of all elements in D_{uo}).

```
Input:  M ∈ Z^{(m+n_c)×(n_uc+n_uo+m+n_c)} and i
while i ≤ min(n_uc + n_uo, m)
   if any M(i...m, i) < 0 then
      Find row j in M(i...m, i) which
        contains a negative element.
      Exchange rows i and j of M
      k := 1
   else
      k := 0
   end if
   if any M(i...m, i) > 0 then
      Find row j in M(i...m, i) which
        contains a positive element.
      Exchange rows i + k and j of M
      Use Algorithm 4 and pivot M(i + k, i)
        to eliminate all negative integers
        in M(i + k...m + n_c, i)
   end if
   if k = 1 then
      Eliminate positive integers in
        M(i...m + n_c, i) with pivot M(i, i)
   end if
   if any M(m+1...m+n_c, i) ≠ 0 then
     FAIL
   end if
   i := i + 1
end while
Output M
```

Lemma 4. A single controller place can possess either an input or an output arc (or neither) to any given plant transition, but it will never contain both, i.e., controllers constructed according to the method outlined in section 1 contain no self-loops.

Proof. The incidence matrix of the invariant forming controller for establishing the constraint $L\mu_p \leq b$ is defined by $D_c = -LD_p$. Positive elements in D_c refer to arcs from plant transitions to the controller, and negative elements refer to arcs from the controller to the plant transitions. There are no allowances for self loops in this controller construction method. The matrix D_c^+ consists of the positive elements in D_c, D_c^- consists of the negative, and all other elements are zero, i.e., there are never any cancellations between D_c^+ and D_c^- used in forming D_c. See [5]. □

A positive element in the matrix LD_{uc} means that the controller would draw an arc to a transition in the un-

```
Algorithm 4 (Column Zeroing).
Input:  M ∈ Z^{(m+n_c)×(n_uc+n_uo+m+n_c)}
  and pivot position (p,j).
i := p+1
while i ≤ m + n_c
   if M(i,j)M(p,j) < 0 then
      while M(i,j) ≠ 0
         if |M(p,j)| > |M(i,j)| then
            d := floor (−M(p,j)/M(i,j))
            if (mod(M(p,j), M(i,j))  =  0)
              then d := d − 1
            end if
            M(p,.) := M(p,.) + dM(i,.)
         else
            d := floor (−M(i,j)/M(p,j))
            M(i,.) := M(i,.) + dM(p,.)
         end if
      end while
   end if
   i := i + 1
end while
Output M
```

controllable portion of the plant. Lemma 4 tells us that this element is not the result of battling input and output arcs. The controller wishes to output to this transition, and it has no reason to receive input from this transition. Algorithm 2 would then be used to eliminate the controller's interaction with the transition all together. But what would happen if row operations were used to change this output transition of the controller into an input transition? The authors know of no situation in which this transformation is necessary or even desirable, however no proof exists at this time to verify this observation.

4 Conclusions

This paper represents ongoing work in the formulation and formalization of a discrete event system control design procedure that relies on simple and efficient linear algebraic techniques. A systematic method for dealing with unobservable, as well as uncontrollable, plant transitions has been presented in section 2 which included new conditions dictating whether or not a controller exists for a particular set of transformed constraints. The usefulness and necessity of transforming both the right and left hand sides of the set of plant constraints have been demonstrated, and two new procedures for automatically generating valid constraint transformations have been presented.

An important question for future research in this area deals with the optimality of controllers that are produced using the procedures described here. Recent work shows a possible relationship between the uniqueness of the controller and the optimality of the control law. Further work is needed in this area before any claims can be made.

References

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

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

[3] P. J. G. Ramadge and W. Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–97, 1989.

[4] John O. Moody, Katerina Yamalidou, Michael D. Lemmon, and Panos J. Antsaklis. Feedback control of Petri nets based on place invariants. In *Proceedings of the 33rd Conference on Decision and Control*, volume 3, pages 3104–3109, Lake Buena Vista, FL, December 1994.

[5] Katerina Yamalidou, John O. Moody, Michael D. Lemmon, and Panos J. Antsaklis. Feedback control of Petri nets based on place invariants. *Automatica*, 32(1):15–28, January 1996.

[6] Alessandro Giua, Frank DiCesare, and Manuel Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 974–979, Chicago, IL, October 1992.

[7] Yong Li and W. Murray Wonham. Control of vector discrete event systems I – the base model. *IEEE Transactions on Automatic Control*, 38(8):1214–1227, August 1993. Correction in IEEE TAC v. 39 no. 8, pg. 1771, Aug. 1994.

[8] Yong Li and W. Murray Wonham. Control of vector discrete event systems II – controller synthesis. *IEEE Transactions on Automatic Control*, 39(3):512–530, March 1994.

[9] John O. Moody, Panos J. Antsaklis, and Michael D. Lemmon. Feedback Petri net control design in the presence of uncontrollable transitions. In *Proceedings of the 34th Conference on Decision and Control*, volume 1, pages 905–906, New Orleans, LA, December 1995.

[10] John O. Moody, Panos J. Antsaklis, and Michael D. Lemmon. Automated design of a Petri net feedback controller for a robotic assembly cell. In *Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, volume 2, pages 117–128, Paris, France, October 1995.