Proceedings of the 33rd Annual Allerton Conference on Communication, Control, and Computing Oct 4-6, 1995 Monticello, IL

Petri Net Supervisors for DES in the Presence of Uncontrollable and Unobservable Transitions[1]

John O. Moody and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556

Abstract

This paper describes a computationally efficient method for synthesizing feedback controllers for plants modeled by Petri nets which may contain uncontrollable transitions. The controller, a Petri net itself, computed using the concept of Petri net place invariants, enforces a set of linear constraints on the plant. It is shown how the original set of plant behavioral constraints can be transformed into a new set that will yield a controller which enforces the original constraints without directly influencing any of the uncontrollable transitions. Previous researchers have shown that this constraint transformation can be performed by solving a linear integer programming problem; the method presented in this paper is simpler computationally, but, depending on the structure of the uncontrollable part of the plant, maximal permissiveness may not be guaranteed.

1 Introduction

A methodology to automatically derive feedback controllers described by a Petri net was presented in [4]. It is assumed that the DES plant is described by an ordinary Petri net. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net. The resulting controllers are identical to the monitors [2] of Giua et al. and consist only of places and arcs. The controller's size is proportional to the number of constraints.

This approach has been recently extended to apply to Petri nets which contain uncontrollable and unobservable transitions, the firing of which cannot be controlled or cannot be observed respectively. This extension is also based on place invariants and it is again easy and transparent to implement with excellent numerical properties. These results on Petri nets with uncontrollable transitions complement and are compared with results on uncontrollable vector DES recently reported in the literature [3].

This invariants based approach to design of Petri net feedback controllers has a number of advantages: (a) the design method is transparent as it is based on the concept of place invariants, (b) the resulting controller and consequently the overall controlled system are described by ordinary Petri nets where a variety of tools for analysis both graphical and algebraic are available. Verification of the design is therefore rather straightforward. (c) The design method has excellent numerical properties. Although it is based on the concept of invariants, it is not actually necessary to calculate any of these invariants. The controller design involves only a multiplication of the incidence matrix by a vector representing the constraints to be imposed on the system. This make this control design approach particularly appealing in control reconfiguration applications where because of a failure the controller must be redesigned on line.

The paper is structured as follows. The controller synthesis method [2,4] for plants with controllable transitions is reviewed in section 2. The new approach for transforming constraints in the face of uncontrollable transitions and generating the corresponding controller is presented in section 3. Concluding remarks are given in section 4.

[1]The research for this paper has been supported in part by the Center for Applied Mathematics at the University of Notre Dame.

J.O. Moody and P.J. Antsaklis, "Petri Net Supervisors for DES in the Presence of Uncontrollable and Unobservable Transitions," Proceedings of the 33rd Annual Allerton Conference on Communications, Control and Computing, pp. 176-185, Allerton House, Monticello, IL, October 4-6, 1995.

2 Automatic Controller Synthesis from Plant Constraints

The system to be controlled is modeled by a Petri net with n places and m transitions and is known as the plant or *process net*. The incidence matrix of the process net is D_p. It is assumed that all the enabled transitions can fire. It is possible that the process net will violate certain constraints placed on its behavior, thus the need for control. The *controller net* is a Petri net with incidence matrix D_c made up of the process net's transitions and a separate set of places. The *controlled system* or *controlled net* is the Petri net with incidence matrix D made up of both the original process net and the added controller. The control goal is to force the process to obey constraints of the following form

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

where μ_p is the marking vector of the Petri net modeling the process, L is an $n_c \times n$ integer matrix, b is an n_c dimensional integer vector and n_c is the number of constraints. Note that the inequality is with respect to the individual elements of the two vectors $L\mu_p$ and b and can be thought of as the logical conjunction of the individual "less than or equal to" constraints. This definition will be used throughout this paper whenever vectors appear on either side of an inequality sign.

Suppose we wish to enforce the single constraint $\mu_1 + \mu_2 \leq 1$ which means that at most one of the two places p_1 and p_2 can be marked, or, in other words, both places cannot be marked at the same time. This inequality constraint can be transformed into an equality by introducing an external Petri net controller which contains a place which represents a nonnegative "slack variable" μ_c. The constraint then becomes $\mu_1 + \mu_2 + \mu_c = 1$ or, in general

$$L\mu_p + \mu_c = b \tag{2}$$

where μ_c is an n_c dimensional integer vector which represents the marking of the controller places. Note that $\mu_c \geq 0$ because the number of tokens in a place can not become negative; thus equation (2) implies inequality (1). The controller places insure that the weighted sums of tokens in the process net's places are always less than or equal to the elements of b. The places which maintain the inequality constraints are part of a separate net called the controller net. The structure of the controller net will be computed by observing that the introduction of the slack variables forces a set of place invariants on the overall controlled system defined by equation (2).

A place invariant is defined as every integer vector x which satisfies

$$x^T\mu = x^T\mu_0 \text{ (a constant)} \tag{3}$$

where μ_0 is the net's initial marking, and μ represents any subsequent marking. Equation (3) means that the weighted sum of the tokens in the places of the invariant remains constant at all markings and this sum is determined by the initial marking of the Petri net. The place invariants of a net are elements of the kernel of the net's incidence matrix, i.e., they can be computed by finding integer solutions to

$$x^T D = 0 \tag{4}$$

where D is an $n \times m$ incidence matrix with n being the number of places and m the number of transitions.

The matrix D_c contains the arcs that connect the controller places to the transitions of the process net. Let $\mathbb{Z}$ be the set of integers. The incidence matrix $D \in \mathbb{Z}^{(n+n_c)\times m}$ of the closed loop system is given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \tag{5}$$

and the marking vector $\mu \in \mathbb{Z}^{n+n_c}$ and initial marking μ_0 are given by

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \qquad \mu_0 = \begin{bmatrix} \mu_{p_0} \\ \mu_{c_0} \end{bmatrix} \tag{6}$$

Note that equation (2) is in the form of (3), thus the invariants defined by equation (2) on the system (5), (6) must satisfy equation (4).

$$\begin{aligned} X^T D = [L \;\; I] \begin{bmatrix} D_p \\ D_c \end{bmatrix} &= 0 \\ LD_p + D_c &= 0 \end{aligned} \tag{7}$$

where I is an $n_c \times n_c$ identity matrix since the coefficients of the slack variables in equation (2) are all equal to 1.

Proposition 1. The Petri net controller, $D_c \in \mathbb{Z}^{n_c \times m}$ with initial marking μ_{c_0}, which enforces constraints (1) when included in the closed loop system (5) with marking (6) is defined by

$$D_c = -LD_p \tag{8}$$

with initial marking

$$\mu_{c_0} = b - L\mu_{p_0} \tag{9}$$

assuming that the transitions with input arcs from D_c are controllable.

Proof. Equation (8) is simply the solution of equation (7) which forces $\begin{bmatrix} L & I \end{bmatrix}$ to be invariants of the closed loop system. Since $\begin{bmatrix} L & I \end{bmatrix}$ are invariants we known that $L\mu_p + \mu_c = L\mu_{p_0} + \mu_{c_0}$ and from equation (9), $L\mu_{p_0} + \mu_{c_0} = b$. The marking of a place is always greater than or equal to zero, therefore

$$\begin{aligned} L\mu_p + \mu_c &= b \\ L\mu_p &\leq b \end{aligned}$$

□

Remark. Proposition 1 creates a controller which will enable and inhibit various transitions in the plant. If any of these transitions are uncontrollable then the controller defined by this method is invalid. The next section shows how a transformation of the constraints can be performed so that the uncontrollable transitions in the net receive no input arcs from the controller places.

Remark. Since our method admits the structure of the process net as well as a set of specifications, it can control transitions that participate in self-loops in the process net. This is because the constraints on these transitions are part of the specifications. Note that when an element of D_c is zero, there are no arcs at all connecting the given place and transition, i.e., there are no canceling self-loops in the controller structure.

Remark. The controller defined by proposition 1 is maximally permissive, assuming controllable transitions, in that it will never disable a transition that would directly violate the constraints if fired. The proof of this result is given in [4].

3 Handling Uncontrollable Transitions

Consider the situation where the controller is not allowed to influence certain transitions in the plant Petri net. These transitions are called uncontrollable. It is illegal for the Petri net controller to include an arc from one of the controller places to any of these uncontrollable plant transitions.

Let D_u be the incidence matrix of the uncontrollable portion of the process net. D_u is composed of the columns of D_p which correspond to the uncontrollable transitions. Recall that, assuming no self loops, positive elements in an incidence matrix refer to arcs from transitions to places, and negative elements refer to arcs from places to transitions. $D_u \in \mathbb{Z}^{m \times n_u}$ where n_u is the number of uncontrollable transitions. Given a set of constraints, $L\mu \leq b$, the Petri net controller given by $D_c = -LD_p$ violates the uncontrollability constraint if LD_u contains any elements greater than zero. The uncontrollability constraint dictates that we can not draw any arcs from the controller places to the transitions, and the portion of the controller corresponding to the uncontrollable transitions is given by $-LD_u$, thus the elements of LD_u must all be less than or equal to zero.

3.1 Controller Computation

Suppose we are unable to meet the uncontrollability constraint and have positive values in the matrix LD_u. It is necessary to transform the constraint vector L such that the original constraint of $L\mu_p \leq b$ is still maintained, while obeying the uncontrollability constraint. A proposition is given later in this section which shows how to construct a controller which meets both of these conditions. The proposition is supported by the lemma below.

Lemma 2. If

$$L'\mu_p \leq b' \tag{10}$$

where

$$\begin{aligned} L' &= R_1 + R_2 L && (11) \\ b' &= R_2(b+1) - 1 && (12) \end{aligned}$$

$R_1 \in \mathbb{Z}^{n_c \times m}$, $R_2 \in \mathbb{Z}^{n_c \times n_c}$, 1 is an n_c dimensional vector of 1's and

$$R_1\mu_p \geq 0 \qquad \text{for all possible } \mu_p \tag{13}$$

$$R_2 \text{ is a positive definite diagonal matrix} \tag{14}$$

then $L\mu_p \leq b$.

Proof. The transformed constraint is

$$\begin{aligned} L'\mu_p &\leq b' \\ (R_1 + R_2 L)\mu_p &\leq R_2(b+1) - 1 \end{aligned}$$

because all of the elements are integers, the inequality can be transformed into a strict inequality:

$$(R_1 + R_2 L)\mu_p < R_2(b+1)$$

and because R_2 is diagonal and positive definite,

$$R_2^{-1} R_1 \mu_p + L\mu_p < b + 1$$

Assumptions (13) and (14) imply that all elements of the vector $R_2^{-1}R_1\mu_p$ must be greater than or equal to zero, therefore

$$L\mu_p < b + 1$$

and, once again because the inequality deals strictly with integers, we have

$$L\mu_p \leq b$$

□

According to proposition 1, the incidence matrix of the controller which will enforce (10) is given by $D_c = -L'D_p$. In order to meet the uncontrollability constraint we need $L'D_u \leq 0$ which will insure that the controller contains no arcs leading to the uncontrollable transitions in the plant. Then

$$\begin{aligned} R_1D_u + R_2LD_u &\leq 0 \qquad \text{or} \\ \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_u \\ LD_u \end{bmatrix} &\leq 0 \end{aligned} \tag{15}$$

Note that if $L'D_u = 0$ then the rows of the matrix $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ lie within the kernel of $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$.

A trivial solution to inequality (15) is given by $\begin{bmatrix} R_1 & R_2 \end{bmatrix} = \begin{bmatrix} -L & I \end{bmatrix}$. However this choice for R_1 will either violate assumption (13) or dictate an impossible control task. If $L\mu$ has both positive and negative elements then we can not say that $R_1\mu = -L\mu$ is always greater than zero and assumption (13) is violated. If $L\mu$ is always positive or zero then $R_1\mu = -L\mu$ violates assumption (13) for arbitrary μ. If $L\mu$ is always negative or zero and $b \geq 0$ then the constraints will always be met by the plant without control, and if $L\mu$ is always negative or zero and $b \leq 0$ then a choice of $R_1\mu = -L\mu$ will create a transformed constraint of $0 \leq b$ which is meaningless since b is a constant vector.

Lemma 2 may now be combined with proposition 1 to synthesize a Petri net controller for a plant with uncontrollable transitions.

Proposition 3. Given a plant Petri net with incidence matrix D_p, a set of uncontrollable transitions, a set of linear constraints $L\mu \leq b$ on the net marking, and R_1 and R_2 which meet assumptions (13) and (14) and satisfy inequality (15), then there exists a controller with incidence matrix

$$D_c = -(R_1 + R_2L)D_p \tag{16}$$

such that if

$$\mu_{c_0} = R_2(b+1) - 1 - (R_1 + R_2L)\mu_{p_0} \tag{17}$$

is a valid initial marking, i.e. $\mu_{c_0} \geq 0$, then the constraint $L\mu_p \leq b$ will be satisfied by the closed loop system. This controller contains no input arcs to any of the uncontrollable transitions.

Proof. Note that, according to proposition 1, equations (16) and (17) define a controller that enforces the constraint $L'\mu \leq b'$ where $L' = R_1 + R_2L$ and $b' = R_2(b+1) - 1$. Lemma 2 shows that if assumptions (13) and (14) are met then a controller which enforces a particular constraint $L'\mu \leq b'$ will also enforce the constraint $L\mu \leq b$. Because R_1 and R_2 satisfy inequality (15), no controller arcs are drawn to the uncontrollable transitions.
□

Remark. If any element of μ_{c_0} calculated according to equation (17) is negative then the corresponding constraint is impossible to enforce. This is the situation where only trivial solutions to the constraint transformation exist because the uncontrollable portion of the plant is incompatible with a particular control goal.

The usefulness of proposition 3 lies in whether or not it is possible to find R_1 and R_2 which meet the necessary assumptions. To meet assumption (13), it is sufficient to assume that all of the elements of R_1 are nonnegative, since the elements of μ_p are nonnegative by definition. In general, for unbounded μ_p, it is necessary that all of the elements of R_1 be nonnegative, however if bounds on μ_p are known, then it is possible to generate valid R_1 vectors which contain some negative elements. If R_1 and R_2 which satisfy (13) and (14) do exist, then they can be found by performing row operations on $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$. Row operations act as premultiplications of a matrix, just as $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ premultiplies $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$ in inequality (15). R_1 and R_2 can be found by finding a set of row operations which do not involve premultiplication of any row by a negative number and which force the LD_u portion of the matrix to contain all zero or negative elements. This operation is relatively easy to perform. Note that assumption (14), which requires R_2 to be a positive definite matrix, is not restrictive. This matrix simply represents the premultiplication coefficients of the rows of the LD_u portion of the matrix undergoing row operations. We can assume this matrix is diagonal because LD_u is linearly dependent with D_u, i.e., we will never need to take linear combinations of the rows in LD_u. We can also assume that the diagonal elements are positive since, if negative numbers are required, they can be accounted for by R_1, which still needs to meet assumption (13).

It is possible that uncontrollable transitions in a plant might make a particular constraint impossible to realize. In this case it may still be possible to find R_1 and R_2 such that they meet the assumptions, however the transformed constraint $L'\mu_p \leq b'$ will be trivial. For example, consider the Petri net in figure 1. Suppose that we wish to limit the number of tokens that enter p_2, i.e., the untransformed constraint is $\mu_2 \leq b$. If the single transition is uncontrollable, then we will obtain a transformed constraint of $\mu_1 + \mu_2 \leq b$, which is already the case if there are b or fewer tokens in the net and is impossible if the net starts with more than b tokens.

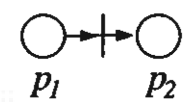

Figure 1: A Petri net that yields a trivial constraint transformation.

3.2 Example

We now provide a simple example in order to illustrate the concepts that have been covered above. The example plant is partially based on the model of an "unreliable machine" from [1]. The machine is used to process parts from an input queue, completed parts are moved to an output queue. The machine is considered unreliable because it is possible that it may break down and damage a part during operation. This behavior is captured in the plant model. Damaged parts are moved to a separate queue from the queue for successfully completed parts. The Petri net model of the plant is shown in figure 2, and a description of the various places and transitions is given in table 1.

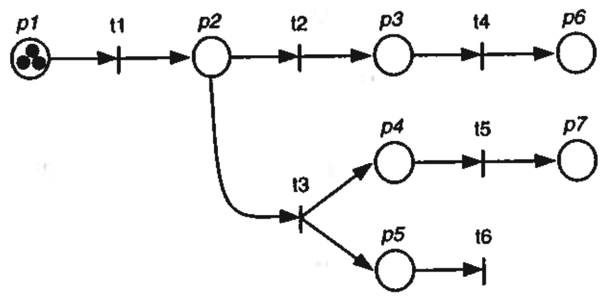

Figure 2: Petri net model of an uncontrolled unreliable machine.

Places

p_1	Input queue - Number of tokens = parts remaining
p_2	Machine is "up and busy," part is being processed
p_3	Part is waiting for transfer to completed parts queue
p_4	Part is waiting for transfer to damaged parts queue
p_5	Machine is waiting to be repaired
p_6	Completed parts queue
p_7	Damaged parts queue

Transitions

t_1	Part moves from input queue to machine
t_2	*Uncontrollable:* Part processing is complete
t_3	*Uncontrollable:* Machine fails, part is damaged
t_4	Part moves to completed parts queue
t_5	Part moves to damaged parts queue
t_6	Machine is repaired

Table 1: Place and transition descriptions for the Petri net of figure 2.

The plant model has two uncontrollable transitions, t_2 and t_3. Transition t_3 represents machine break down and so obviously can not be controlled. Transition t_2 is considered uncontrollable because the controller can not force the machine to instantly finish a part that is not yet completed, nor does it direct the machine to stop working on an unfinished part. The transition is labeled uncontrollable in order to prevent a control design from attempting either of these two actions.

The Petri net model of the plant has the following incidence matrix and marking vector.

$$D_p = \left[\begin{array}{cccccc} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \underbrace{0 \quad 0}_{D_u} & & 0 & 0 & 1 & 0 \end{array}\right] \qquad \mu_p = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{bmatrix} \tag{18}$$

The initial conditions are $\mu_{p_0} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$.

If the machine is broken, we do not want to load a new part until repairs have been completed. This means that places p_2 and p_5 should contain at most one token:

$$\mu_2 + \mu_5 \leq 1 \tag{19}$$

Parts waiting to be transferred to a storage queue, whether completed or damaged, wait in the same position on the machine. The Petri net model uses two places, p_3 and p_4, to represent waiting parts, because there are two different destinations. In order to prevent conflict, the second constraint is

$$\mu_3 + \mu_4 \leq 1 \tag{20}$$

Using the matrix form of constraint (1) we have

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b} \tag{21}$$

First we must check the uncontrollability condition.

$$LD_u = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$$

We need all of the elements of LD_u to be less than or equal to zero if we are to avoid using uncontrollable transitions. There is no problem with the first row, but a transformation will have to be found to eliminate the 1's in the second row. This can be done by applying row operations from the matrix D_u to eliminate the positive elements in the second row of LD_u.

Because constraint (19) required no transformation, the first row of R_1 will be all zeros. A row operation involving the addition of the second row of the D_u matrix was required to transform constraint (20), thus the second row of R_1 will be all zeros with a one in the second column. It was not necessary to premultiply either constraint, thus R_2 will be an identity matrix.

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We now apply equations (11) and (12) to find the transformed constraints represented by L' and b'.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}}_{L'} \mu_p \leq \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{b'}$$

The controller is the calculated using equations (16) and (17).

$$D_c = -L'D_p = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\mu_{c_0} = b' - L'\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The controlled net is shown in figure 3. The constraint logic is enforced and no input arcs are drawn to the uncontrollable transitions.

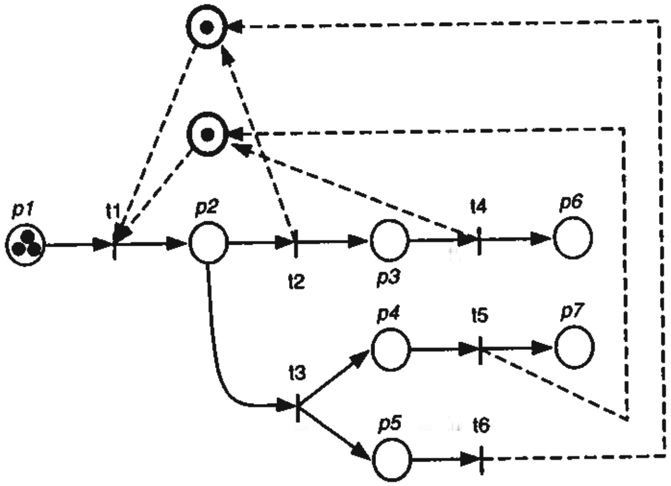

Figure 3: The controlled unreliable machine.

3.3 Discussion

An extensive look at many of the issues central to this research can be found in the work of Li and Wonham [3]. These authors show that optimal, or maximally permissive, control actions which account for uncontrollable transitions can be found by repeated applications of a linear integer programming problem (LIP), assuming that valid control actions actually exist and that the uncontrollable portion of the net contains no loops. They also give sufficient conditions under which the solution to the LIP has a closed form expression. These conditions place a certain tree structure on the uncontrollable portion of the net. When this tree structure is further limited, Li and Wonham are able to prove that the optimal control law which insures $L\mu_p \leq b$ can be written $C\mu_p \leq d$. This is the case where it is possible to represent the action of the optimal control law with ordinary Petri nets. In this situation, it is possible to find R_1 and R_2 by performing row operations on $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$ which is much more desirable, computationally, than analytically solving an LIP. However the tree structure assumed by Li and Wonham is only sufficient, not necessary, for example, the structure of the uncontrollable part of the plant in section 3.2 does not conform to Li and Wonham's "type 2 tree structure," however an optimal solution was found and implemented using an ordinary Petri net controller. There are also cases where, following the procedures presented above, suboptimal Petri net controllers

may be derived. These suboptimal controllers may be sufficient for many tasks, depending on the application. Further exploration and codification of these issues is one of the main goals of this ongoing research.

4 Conclusions

This paper has presented a particularly simple method for constructing feedback controllers for untimed Petri nets, even in the face of uncontrollable plant transitions. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net, and that simple row operations on a matrix containing the uncontrollable columns of the plant incidence matrix can be used to eliminate controller use of illegal transitions.

The significance of this particular approach to Petri net controller design is that the control net can be computed very efficiently, thus the method shows promise for controlling large, complex systems, or for recomputing the control law online due to some plant failure.

Besides the problem of uncontrollable plant transitions, there are also situations where it is not possible to observe certain transitions. For example, it may be too difficult or costly to place a sensor in a certain part of a plant, or a sensor may fail which might require the fast online computation of a new control law. Because of a duality in the mathematics between uncontrollable and unobservable transitions, this research extends itself naturally to the problem of unobservable transitions.

References

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

[2] A. Giua, F. DiCesare, and M. Silva, "Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions", *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, Chicago, IL, pp. 974–979, October 1992.

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

[4] J. O. Moody, K. Yamalidou, M. Lemmon, and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants", *Proceedings of the 33rd Conference on Decision and Control*, Vol. 3, pp 3104–3109, Lake Buena Vista, FL, December 1994.

[5] 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, January 1989.

[6] W. Reisig, *Petri Nets*, Springer-Verlag, 1985.

[7] W. M. Wonham and P. J. G. Ramadge, "On the Supremal Controllable Sublanguage of a Given Language", *SIAM J. Control Optim.*, vol. 25, no. 3 pp 637-659, 1987.