Copyright © 1996 IFAC
13th Triennial World Congress, San Francisco, USA
Vol B

Ia-02 6

PETRI NET FEEDBACK CONTROLLER DESIGN FOR A MANUFACTURING SYSTEM[1]

John O. Moody, Panos J. Antsaklis, and Michael D. Lemmon

Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556
USA

Abstract. This paper describes a computationally efficient method for synthesising feedback controllers for plants modeled by Petri nets while illustrating the method with a running example of a manufacturing system. The controller, a Petri net itself, computed using the concept of Petri net place invariants, enforces a set of linear constraints on the plant. Computationally efficient and automatically derived techniques are given to deal with uncontrollable transitions in the plant and the on-line recomputation of the controller due to sensor failures.

Keywords. Petri Nets, Discrete Event Systems, Manufacturing Systems

1. INTRODUCTION

A Petri net supervisor which enforces a set of linear constraints on the marking behavior of a Petri net plant can be efficiently computed using a simple matrix equation. It has been shown by Moody *et al.* (1994) and Yamalidou *et al.* (1996) how to derive these controllers using the concept of Petri net place invariants; see Murata (1989), Peterson (1981), and Reisig (1985) for descriptions of the algebraic representation and properties of Petri nets. Supervisors constructed in this manner are identical to the monitors of Giua *et al.* (1992). This supervisory control method (Ramadge and Wonham, 1989; Wonham and Ramadge, 1987) is a powerful means of realizing these linear constraints because it is simple to calculate, and the Petri net structure of the solution makes the controller easy to implement. Unfortunately this method can not be directly applied to plants that contain uncontrollable or unobservable transitions, i.e. plant transitions that either can not be prevented from firing when enabled, or that can not be detected when fired. When the plant contains these kinds of transitions, it is often possible to implement a Petri net controller, but this controller can no longer be calculated directly from the control goals. In order to deal with uncontrollability and unobservability, it is first necessary to transform the original set of plant constraints into a set which accounts for the problem transitions. Once this transformation has been computed, controllers can be automatically generated from the set of transformed constraints. Li and Wonham (1993; 1994) have shown how the transformation can be performed by symbolically solving a linear integer programming problem. An alternate approach is proposed here, showing how the transformation can be performed in a computationally efficient manner using matrix row operations. Possible uses for the types of plant constraints which can be implemented, as well as the control method itself, are illustrated with the example of a manufacturing system.

The paper is organised as follows. Section 2 outlines the methods for modifying plant constraints to account for

[1] Partially supported by the Center for Applied Mathematics at the University of Notre Dame, and the Army Research Office (DAAH04-95-0600).

uncontrollable plant transitions. A piston rod assembly robotic manufacturing cell plant model is presented, and an appropriate supervisor is computed for it in section 3. The idea of uncontrollable transitions is extended to unobservable transitions in section 4 where a new controller is designed to account for a sensor failure in the assembly cell. Section 5 contains concluding remarks.

2. UNCONTROLLABLE TRANSITIONS

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 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 $D = \begin{bmatrix} D_p \\ D_c \end{bmatrix}$ made up of both the original process net and the added controller. The goal is to force the process to obey n_c linear constraints of the 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.

A maximally permissive Petri net controller (Giua *et al.*, 1992; Moody *et al.*, 1994) which enforces constraints (1) when included in the closed loop system D is defined by the incidence matrix

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

and the initial marking

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

assuming that the transitions with input arcs from D_c are controllable and that the constraints are not contradictory, i.e. $\mu_{c_0} \geq 0$.

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_{uc} be the incidence matrix of the uncontrollable portion of the process net. D_{uc} 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. $D_{uc} \in Z^{m \times n_u}$ 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_{uc} 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 $-LD_{uc}$, thus the elements of LD_{uc} must all be less than or equal to zero.

Suppose we are unable to meet the uncontrollability constraint, i.e. we have positive values in the matrix LD_{uc}. 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. The constraint transformation will take the form

$$L'\mu_p \leq b' \tag{4}$$

where

$$L' = R_1 + R_2 L \tag{5}$$

$$b' = R_2(b+1) - 1 \tag{6}$$

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

$$R_1\mu_p \geq 0 \qquad \text{for all possible } \mu_p \tag{7}$$

$$R_2 \text{ is a positive definite diagonal matrix} \tag{8}$$

It is shown in (Moody *et al.*, 1995) that any controller which enforces $L'\mu_p \leq b'$ will also enforce $L\mu_p \leq b$.

In order to meet the uncontrollability constraint we need $L'D_{uc} \leq 0$ which will insure that the controller contains no arcs leading to the uncontrollable transitions in the plant. We need

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_{uc} \\ LD_{uc} \end{bmatrix} \leq 0 \tag{9}$$

Thus if R_1 and R_2 can be found which satisfy inequality (9) and satisfy assumptions (7) and (8) then a controller can be synthesised which enforces $L\mu_p \leq b$ and does not contain any input arcs to the uncontrollable transitions

J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Petri Net Feedback Controller Design for a Manufacturing System," P roc o f t he I FAC 1 3th T riennial W orld C ongres s , Vol. B, pp. 67-72, San Francisco, CA, July 1-5, 1996.

in the plant. In general R_1 and R_2 are found by performing row operations on $\begin{bmatrix} D_{uc} \\ LD_{uc} \end{bmatrix}$. This procedure is demonstrated in the following sections.

3. PISTON ROD ROBOTIC ASSEMBLY CELL

The automatic control concepts discussed above are illustrated here using the example of a piston rod assembly cell which is taken from chapter 8 of (Desrochers and Al-Jaar, 1995). The Petri net model of the plant 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.

p_1	Work area clear, engine block and crank shaft ready.
p_2	S-380 robot aligns the crank shaft.
p_3	S-380 robot picks up new piston rod and positions it.
p_4	M-1 robot picks up the piston pulling tool.
p_5	M-1 robot positions piston rod and returns pulling tool.
p_6	M-1 robot picks up a cap and positions it on piston rod.
p_7	M-1 robot fasten cap to piston rod using two bolts.

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

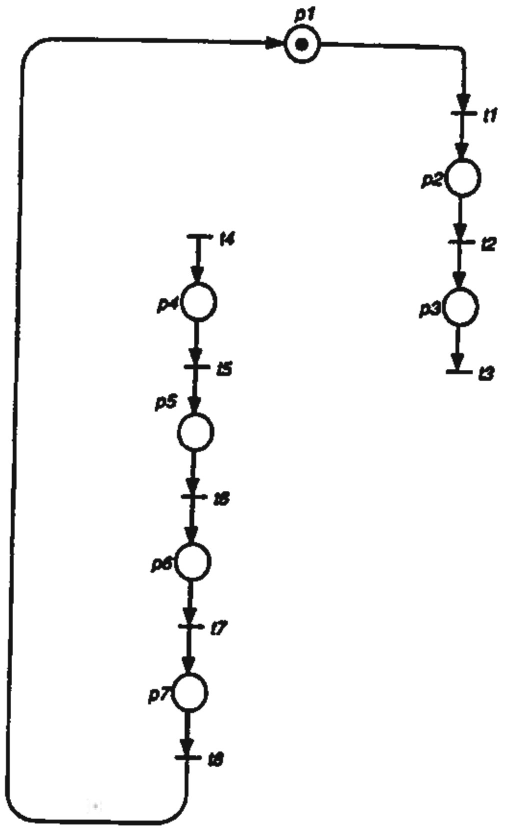

Fig. 1. The base Petri net model for the piston rod robotic assembly cell.

$$D_p = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad \mu_{p0} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (10)$$

There are a number of constraints that must be imposed on this assembly cell in order to properly synchronise the robots and to insure that physical limitations are obeyed. There is only one S-380 and one M-1 robot available for the assembly process, thus only one of each robot can be working at any given time. Places p_2 and p_3 represent activity for the S-380 and places p_4, p_5, p_6, and p_7 represent activity for the M-1, thus

$$\mu_2 + \mu_3 \leq 1 \qquad (11)$$

$$\mu_4 + \mu_5 + \mu_6 + \mu_7 \leq 1 \qquad (12)$$

The plant graph already insures that the S-380 will not begin its task until an engine block and crank shaft are

J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Petri Net Feedback Controller Design for a Manufacturing System," P roc o f t he I FAC 1 3th T riennial W orld C ongres s , Vol. B, pp. 67-72, San Francisco, CA, July 1-5, 1996.

available and the work space is clear, however we need to make sure that the M-1 does not try to pull the piston rod into the engine until the S-380 has finished aligning the crank shaft and readying a new piston rod. In other words, we must insure that transition t_5 does not fire until after tranaition t_3 has fired. A review of the graph structure of the plant net shows that we can write this constraint as

$$\mu_1 + \mu_2 + \mu_3 + \mu_5 + \mu_6 + \mu_7 \leq 1 \qquad (13)$$

Note that this constraint allows the M-1 robot to ready its pulling tool, the task of p_4, at the same time that the S-380 robot is performing one of its tasks.

There are several finite resources that are used in the assembly process. A piston rod is readied at p_3, a pulling tool is required at p_4 and p_5, a cap is required at p_6, and two nuts are used at p_7. The plant controller will need to know if any of these resources is unavailable in order to stall operation until the parts are ready. It is possible to provide the controller with the necessary hooks to manage these resources by associating a constraint with each of the places (or sets of places) that requires the use of a finite resource:

$$\begin{aligned} \mu_3 &\leq 1 & (14)\\ \mu_4 + \mu_5 &\leq 1 & (15)\\ \mu_6 &\leq 1 & (16)\\ \mu_7 &\leq 1 & (17) \end{aligned}$$

One final constraint placed on the assembly cell involves the smooth and uninterrupted operation of the M-1 robot. Perhaps due to the nature of the M-1 robot's dynamics or programming, we would prefer that its operation not be interrupted from the point that it pulls the piston rod into the engine block until it has completed fastening the cap on the piston rod. We can model this constraint by marking transitions t_6, t_7 and t_8 as uncontrollable. Thus we have told the controller that once a token passes into p_5, it can not stall the plant's progress until the token has passed on to places p_6, p_7 and completion. This means that if the controller does want to stall the M-1 robot, it should do it before it starts its primary operation.

Constraints (11) – (17) are now combined so that all of the constraints can be expressed in the single matrix inequality $L\mu_p \leq b$. The controller which enforces these constraints will have seven places (one for each row of L), but first we must insure that our constraints meet the uncontrollability condition. Transitions t_6, t_7, and t_8 are uncontrollable, so the matrix D_{uc} is composed of the last three columns of D_p. The matrix LD_{uc} must be checked for any positive values, which would indicate that the current constraints would violate the uncontrollability conditions.

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

Rows 6 and 7 of LD_{uc} contain positive values, thus it will be necessary to manipulate the sixth and seventh constraints (constraints (16) and (17)) so that their enforcement will not generate a controller with arcs directed toward the uncontrollable transitions. Based on the procedure outlined in section 2, the offending rows of LD_{uc} can be eliminated by adding rows from D_{uc}. Keeping track of the row operations performed will yield the matrices R_1 and R_2 which will be used to generate the transformed constraint matrix L'. The row operations are as follows.

$$\begin{array}{rcl} \text{Rows 6 and 7} & = & \begin{bmatrix} 1 & -1 & 0\\ 0 & 1 & -1 \end{bmatrix}\\ \text{of } LD_{uc} & & \\ \text{Add row 5 of } D_{uc} & \Rightarrow & \begin{bmatrix} 0 & -1 & 0\\ 1 & 0 & -1 \end{bmatrix}\\ \text{Add row 6 of } D_{uc} & & \\ \text{Add row 5 of } D_{uc} & \Rightarrow & \begin{bmatrix} 0 & -1 & 0\\ 0 & 0 & -1 \end{bmatrix} \end{array}$$

The row operations performed above correspond to the following transformation in constraints (16) and (17):

$$\begin{aligned} \mu_6 \leq 1 &\Rightarrow \mu_5 + \mu_6 \leq 1 & (18)\\ \mu_7 \leq 1 &\Rightarrow \mu_5 + \mu_6 + \mu_7 \leq 1 & (19)\\ & & (20) \end{aligned}$$

It is now possible to calculate the controller using $D_c = -L'D_p = (R_1 + R_2L)D_p$ where R_1 and R_2 are derived from the row operations above. The controlled net is shown in Fig. 2 with the controller arcs shown as dashed lines and the controller places highlighted in bold. Table 2 describes the meaning of each of the controller places when a token is present within them. Note that place c_3 insures that the M-1 robot will wait to start working on the engine block until the S-380 has completed its task, but it is capable of readying the piston pulling tool while the S-380 is working. Also note that the controller directs no arcs to the uncontrollable transitions, however it still manages to enforce constraints (16) and (17).

J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Petri Net Feedback Controller Design for a Manufacturing System," P roc o f t he I FAC 1 3th T riennial W orld C ongres s , Vol. B, pp. 67-72, San Francisco, CA, July 1-5, 1996.

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 2. Descriptions of the controller places in the Petri net of Fig. 2.

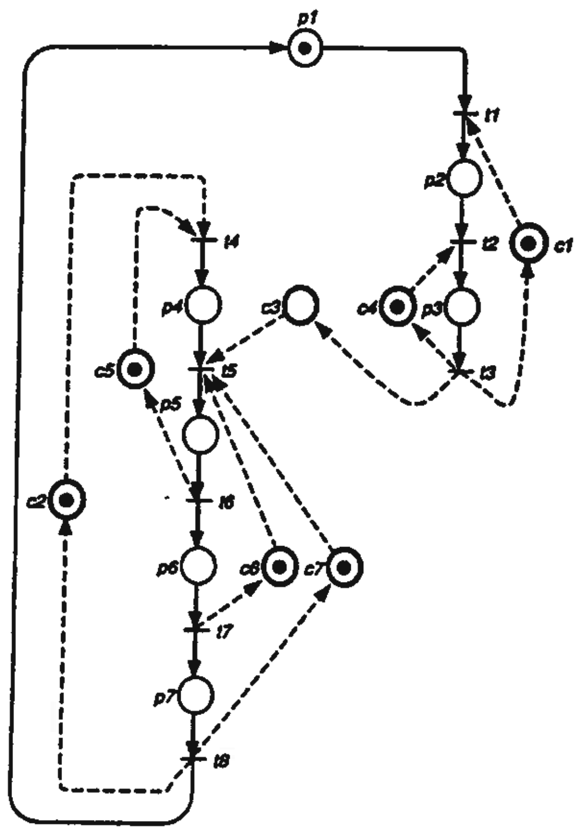

Fig. 2. The assembly cell model with Petri net controller.

4. HANDLING SENSOR FAILURES

Uncontrollable transitions can be used to model the progress of irreversible processes, they can be used to prohibit undesirable control actions, as in section 3, or they can be used to model the failure of an actuator. In the case of an actuator failure a controllable transition is replaced with an uncontrollable one which requires some compensation on the part of the controller or the plant to be shut down. In this section, an analogous concept to uncontrollability, unobservability, will be explored. Unobservable transitions are analogous to uncontrollable transitions both in use and how they are handled analytically. Unobservable transitions may represent events which the controller can not detect, or which are too expensive or inconvenient to detect. An observable transition may be replaced with an unobservable transition in the case of a sensor failure, and this is the situation which will be used to illustrate these concepts here.

The piston rod assembly cell presented in (Desrochers and Al-Jaar, 1995) 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. 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. Unobservable transitions are handled analogously to uncontrollable ones by checking whether $L'D_{uo}$ contains any negative numbers. If $L'D_{uo}$ does contain any negative numbers then we will perform row operations on $L'D_{uo}$ to create a new matrix L'' which obeys the constraints. The new controller incidence matrix will the be given by $D_c = -L''D_p$.

First we observe that the controller needs to be modified to meet the unobservability constraint.

$$L'D_{uo} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

The fifth and sixth rows of $L'D_{uo}$, corresponding to constraints (15) and (18), will need to be modified to eliminate the two -1's. Row operations are performed as in section 3 creating the following transformations in the constraints:

$$\mu_4 + \mu_5 \leq 1 \Rightarrow \mu_4 + \mu_5 + \mu_6 + \mu_7 \leq 1 \qquad (21)$$

$$\mu + 5 + \mu_6 \leq 1 \Rightarrow \mu_5 + \mu_6 + \mu_7 \leq 1 \qquad (22)$$

The new constraint matrix, L'' is then derived from the row operations, and the modified controller is calculated using $Dc = -L''D_p$. The controlled system is shown in Fig. 3. The unobservable transitions do not contain arcs to the controller places. Note that the condition of unobservability has caused places c_2 and c_5 to perform the same function. The same is true for places c_6 and c_7. These places should not be considered redundant because, even though they perform the same functions in the Petri net model, they represent different resources.

J.O. Moody, P.J. Antsaklis and M.D. Lemmon, "Petri Net Feedback Controller Design for a Manufacturing System," P roc o f t he I FAC 1 3th T riennial W orld C ongres s , Vol. B, pp. 67-72, San Francisco, CA, July 1-5, 1996.

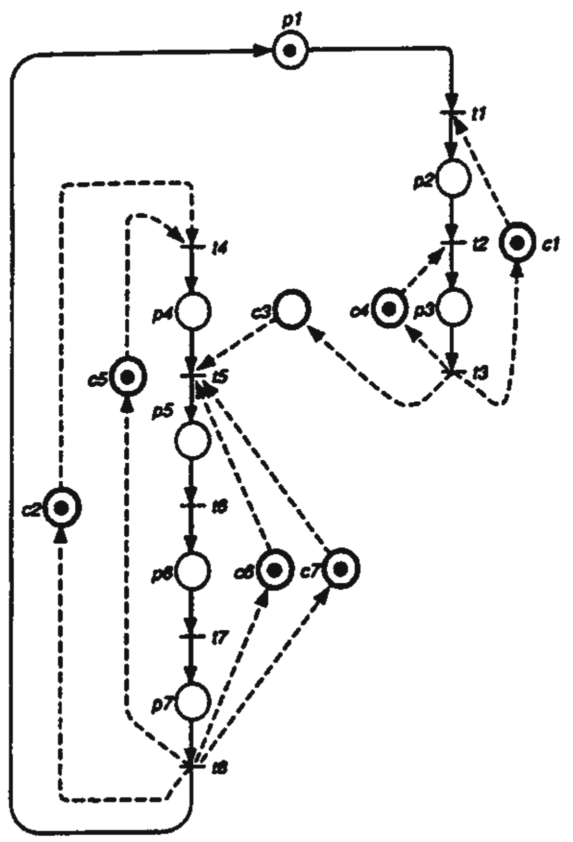

Fig. 3. The assembly cell model with a controller that accounts for a sensor loss making transitions t_5, t_6 and t_7 unobservable.

5. CONCLUSIONS

This paper has presented a particularly simple and computationally efficient method for constructing feedback controllers for untimed Petri nets, even in the face of uncontrollable and unobservable 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/unobservable 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 efficiently and automatically based on the plant constraints. The method shows promise for controlling large, complex systems, or for recomputing control laws online due to some plant failure, such as the loss of a required resource, the break down of an actuator, or the corruption of a sensor.

6. REFERENCES

Desrochers, Alan A. and Robert Y. Al-Jaar (1995). *Applications of Petri Nets in Manufacturing Systems.* IEEE Press. Piscataway, New Jersey.

Giua, Alessandro, Frank DiCesare and Manuel Silva (1992). Generalized mutual exclusion constraints on nets with uncontrollable transitions. In: *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics.* Chicago, IL. pp. 974–979.

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

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

Moody, John O., Katerina Yamalidou, Michael D. Lemmon and Panos J. Antsaklis (1994). Feedback control of Petri nets based on place invariants. In: *Proceedings of the 33rd Conference on Decision and Control.* Vol. 3. Lake Buena Vista, FL. pp. 3104–3109.

Moody, John O., Panos J. Antsaklis and Michael D. Lemmon (1995). Feedback Petri net control design in the presence of uncontrollable transitions. In: *Proceedings of the 34th Conference on Decision and Control.* Vol. 1. New Orleans, LA. pp. 905–906.

Murata, Tadao (1989). Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE* 77(4), 541–580.

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

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

Reisig, Wolfgang (1985). *Petri Nets.* Springer-Verlag. Berlin; New York.

Wonham, W. Murray and P. J. G. Ramadge (1987). On the supremal controllable sublanguage of a given language. *SIAM Journal of Control Optimization* 25(3), 637–659.

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