Feedback Petri Net Control Design
In the Presence of Uncontrollable Transitions

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

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, enforces a set of linear constraints on the plant. The original set of plant behavioral constraints is transformed to yield a controller which enforces the original constraints without influencing any uncontrollable transitions.

1 Introduction
A method is proposed in this paper for transforming a set of linear constraints on the behavior of a plant modeled by a Petri net into an equivalent set which accounts for uncontrollable transitions; see [3] for an alternative methodology based on integer programming. The control goal is to realize a set of behavioral constraints on the plant which enforces the original constraints while obeying the uncontrollability constraint.

Proposition 1. Let

\[ L' \mu_p \leq b' \]

where \( L' \mu_p \) is a vector of \( 1 \)'s, then

\[ L' \mu_p \leq b. \]

Proof. The transformed constraint is \((R_1 + R_2 L)\mu_p \leq R_2(b + 1) - 1.\) Because all of the elements are integers, the inequality can be transformed into a strict inequality:

\[ (R_1 + R_2 L)\mu_p < R_2(b + 1). \]

Assumptions (3) and (4) imply that all elements of the vector \( R_2^{-1} R_1 \mu_p \geq 0, \) therefore \( L' \mu_p < b. \)

Proposition 2. Let a plant Petri net with incidence matrix \( D_p \) be given with a set of uncontrollable transitions, a set of linear constraints \( L' \mu_p \leq b \) on the net marking. Assume \( R_1 \) and \( R_2 \) meet (3) and (4) and let

\[ \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_u & LD_u \end{bmatrix} \leq 0 \]

Then the controller

\[ D_c = -(R_1 + R_2 L)D_p \]

causes all subsequent markings of the closed loop system (1) to satisfy the constraint \( L' \mu_p \leq b. \)

Proof. According to (2), equations (8) and (9) define a controller that enforces the constraint \( L' \mu_p \leq b'. \) Lemma 1 shows that if assumptions (3) and (4) are met then a controller which enforces a particular constraint \( L' \mu_p \leq b' \) will also enforce the constraint \( L' \mu_p \leq b. \) Because \( R_1 \) and \( R_2 \) satisfy inequality (7), no controller arcs are drawn to the uncontrollable transitions. □
The usefulness of proposition 2 lies in whether or not it is possible to find $R_1$ and $R_2$ which meet the necessary assumptions. If $R_1$ and $R_2$ which satisfy (3) and (4) do exist, then they can be found by performing row operations on $\begin{bmatrix} D_u \\ LD_u \end{bmatrix}$.

3 Unreliable Machine Example

The example presented here 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 called unreliable because it is possible that it may break down and damage a part during operation. Damaged parts are moved to a separate queue from the queue for completed parts. The model of the plant is shown in Figure 1; the places are described in Table 1.

![Figure 1: Petri net model of the unreliable machine.](image)

Table 1: Place descriptions for the Petri net of Figure 1.

<table>
<thead>
<tr>
<th>Place</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>Input queue</td>
</tr>
<tr>
<td>$p_2$</td>
<td>Machine is &quot;up and busy&quot; processing part</td>
</tr>
<tr>
<td>$p_3$</td>
<td>Part is waiting for transfer to $p_6$</td>
</tr>
<tr>
<td>$p_4$</td>
<td>Part is waiting for transfer to $p_7$</td>
</tr>
<tr>
<td>$p_5$</td>
<td>Machine is waiting to be repaired</td>
</tr>
<tr>
<td>$p_6$</td>
<td>Completed parts queue</td>
</tr>
<tr>
<td>$p_7$</td>
<td>Damaged parts queue</td>
</tr>
</tbody>
</table>

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.

If the machine is broken, we do not want to load a new part until repairs have been completed. This means that places $p_3$ and $p_5$ should contain at most one token: $\mu_2 + \mu_5 \leq 1$. Parts waiting to be transferred to a storage queue, whether completed or damaged, wait in the same position on the machine. In order to prevent conflict, the second constraint is $\mu_3 + \mu_4 \leq 1$.

A check of the uncontrollability condition shows that $LD_u$ contains positive elements. Row operations are performed to find appropriate values for $R_1$ and $R_2$ which yield the transformed constraints that do not influence the uncontrollable transitions. The controlled net is shown in Figure 2.

![Figure 2: The controlled unreliable machine.](image)

4 Conclusions

This paper has presented a particularly simple method for constructing feedback controllers for untimed Petri nets in the face of uncontrollable plant transitions. The method is based on the idea that 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 systems, or for recomputing the control law online due to a plant failure.

References


