J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.

FA-18 8:50

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

Feedback Control of Petri Nets Based on Place Invariants

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

Abstract

This paper describes a method for constructing a Petri net feedback controller for a discrete event system modeled by a Petri net. The controller enforces a set of linear constraints on the plant and consists of places and arcs. It is computed using the concept of Petri net place invariants. The size of the controller is proportional to the number of constraints which must be satisfied. The method is very attractive computationally, and it makes possible the systematic design of Petri net controllers for complex industrial systems.

1 Introduction

Petri nets are an appropriate tool for the study of discrete-event dynamical systems because of their power and flexibility. They have been used extensively to model and simulate many kinds of systems. Their use in control is somewhat limited and only recently some studies have been conducted towards this direction.

Holloway and Krogh [4] used *controlled Petri nets* to control systems that can be modeled as cyclic controlled marked graphs, which is a special class of Petri nets. They have shown how to synthesize a maximally permissive feedback controller which guarantees that none of the forbidden states will occur. Their method does not require an exhaustive search of the system's state space, is computationally effective and polynomial in the number of forbidden conditions and in the number of places of the net. The drawback is that it is applicable to a limited class of systems.

Yamalidou [11] formulated the control problem of discrete-event chemical processes as a linear optimization problem based on the Petri net model of the process. The control actions which bring the system from its initial state to a desired final state are computed over a time horizon, while a set of constraints are satisfied and a cost function is minimized. The constraints are written as Boolean expressions which are then transformed into sets of linear inequalities.

Boissel [1] used simulated annealing to compute a Petri net controller for a discrete-event system modeled by a Petri net. The method can be applied to any system modeled by uncolored Petri nets and can also handle time. Although successful in producing a maximally permissive optimal controller, the method is computationally unattractive for large systems since it involves the construction of the reachability tree several times during the execution of the algorithm.

More recently Li and Wonham [6] have extended the work of Ramadge and Wonham on discrete event system (DES) control, see [9, 7], to vector DES's (Petri nets). The method given in [6] shows how to reduce the problem of controller construction to a linear integer programming problem under certain assumptions about the plant structure. The method is used to enforce linear constraints, of the type described in this paper, and is extended to handle certain formal language realization problems. Giua and DiCesare [2] have also done recent work on language control and realization with Petri nets. An informative study of Petri net control issues can be found in [3].

The method presented here computes a Petri net controller for a discrete-event system modeled by an untimed Petri net and is based on the net's place invariants. The controller consists of places which are connected to the transitions of the process Petri net in such a way that it is guaranteed that the system does not enter a forbidden state. The combined process/controller net possesses the necessary place invariants to insure that the set of constraints is not violated.

The controller is not necessarily optimal in size, but it is easily computed (involving only matrix multiplications) and its size is proportional to the number of constraints of the process. The controller is maximally permissive in that it forces the set of constraints to be obeyed, while allowing any action that is not directly or indirectly forbidden by the constraints. If the constraints on a net's performance are written in terms of the firing vector, then there are situations in which the maximal permissiveness of the control method can only be guaranteed if the net is safe [10].

The paper is structured as follows. First the theory concerning the place invariants of Petri nets is briefly discussed in section 2. Then the method itself is presented in section 3. Different kinds of constraints are discussed in section 4. Examples are used to illustrate the method in section 5. Conclusions and further research directions are given in section 6.

2 Place Invariants

One of the structural properties of Petri nets, i.e. properties that depend only on the topological structure of the Petri net and not on the net's initial marking, are the net invariants. Here we are interested in *place invariants*;

0-7803-1968-0/94$4.00©1994 IEEE

J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.

Petri nets may also contain transition invariants, see [8].

Place invariants are sets of places whose token count remains constant. They are represented by an n-column vector x, where n is the number of places of the Petri net, whose non-zero entries correspond to the places that belong to the particular invariant and zeros everywhere else. A place invariant is defined as every integer vector x which satisfies

$$\mu^T x = \mu_0^T x \tag{1}$$

where μ_0 is the net's initial marking, and μ represents any subsequent marking. Equation (1) 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 can be computed by finding the integer solutions to

$$x^T D = 0 \tag{2}$$

where D is the $n \times m$ composite change matrix of the Petri net with n being the number of places and m the number of transitions of the net. It is easily shown that every linear combination of place invariants is also a place invariant for the net.

Invariants are important means for analyzing Petri nets since they allow for the net's structure to be investigated independently of any dynamic process [5]. Invariant analysis can be performed on local subnets without considering the whole system and can be used for model verification.

3 Description of the Method

The system to be controlled is modeled by a Petri net with n places and m transitions. The control goal is to force the process to obey constraints of the following form

$$l_1 \mu_i + l_2 \mu_j \leq b_1 \tag{3}$$

where μ_i and μ_j are the markings of places p_i and p_j of the process net, and l_1, l_2, and b_1 are integer constants. If, for example, $l_1 = l_2 = b_1 = 1$, then equation (3) means that at most one of the two places p_i and p_j 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 a *slack variable* μ_s into it. The constraint then becomes

$$l_1 \mu_i + l_2 \mu_j + \mu_s = b_1 \tag{4}$$

The slack variable in this case represents a new place p_s which holds the extra tokens required to meet the equality. It insures that the weighted sum of tokens in places p_i and p_j is always less than or equal to b_1. This place belongs to the *controller net*. The structure of this net will be computed by observing that the introduction of the slack variable introduces a place invariant for the overall *controlled* system defined by equation (4). It is obvious that there will be as many controller places as there are constraints of the type (3), so the size of the controller is proportional to the number of constraints of type (3).

Since a new place has been added to the net, the composite change matrix D of the controlled system is the original $n \times m$ matrix D_p of the system increased by a row corresponding to the place introduced by the slack variable. This new row belongs to the composite change matrix of the controller, called D_c. The arcs connecting the controller place to the original Petri net of the system will be computed by the place invariant equation (2) where the unknowns are the elements of the new row of matrix D while the vector x_i is the place invariant defined by equation (4). These computations are described below.

First note that the problem can be stated in general, as follows. All constraints of type (3) can be grouped and written in matrix form as

$$L\mu_p \leq b \tag{5}$$

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 $n_c \times 1$ integer vector and n_c is the number of constraints of type (3). Note that the inequality is with respect to the individual elements of the two vectors $L\mu_p$ and b.

Similarly all place invariant equations of type (4), generated after the introduction of the slack variables, can be grouped in matrix form as follows

$$L\mu_p + \mu_c = b \tag{6}$$

where μ_c is an $n_c \times 1$ integer vector which represents the marking of the controller places.

Each place invariant defined by equation. (6) must satisfy equation (2):

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

where I is an $n_c \times n_c$ identity matrix since the coefficients of the slack variables in the constraints are all equal to 1. The matrix D_c contains the arcs that connect the controller places to the transitions of the process net. So, given the Petri net model of the process (D_p) and the constraints that the process must satisfy (L and b), the Petri net controller (D_c) is defined by

$$D_c = -LD_p \tag{7}$$

The initial marking of the controller Petri net should also be calculated. The initial marking of the controller places μ_{c_0} must be such that the place invariant equations (6) are satisfied and depends on the initial marking of the places of the process Petri net which participate in the place invariants. Given equation (1), equation (6) can be written for the initial marking vector:

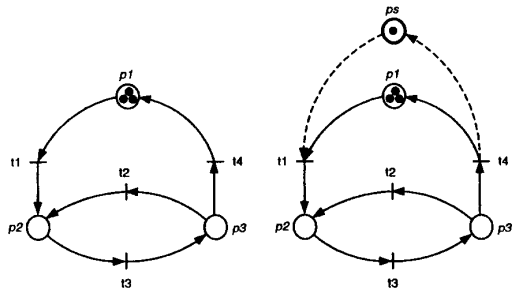
$$\begin{aligned} L\mu_{p_0} + \mu_{c_0} &= b \\ \mu_{c_0} &= b - L\mu_{p_0} \end{aligned} \tag{8}$$

3.1 Example Controller Construction

As an example consider the simple Petri net of figure 1a. The composite change matrix of this net is

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

J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.



a) Process Net b) Controlled Net

Figure 1: Petri nets for the example of section 3.1

while its initial marking is

$$\mu_{p_0} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$$

D_p is of rank 2, thus it has one place invariant which includes the entire net, i.e., $x^T D_p = 0$ where $x^T = [1\ 1\ 1]$. The objective is to control the net so that places p_2 and p_3 never contain more than one token, i.e. we wish to enforce the constraint

$$\mu_2 + \mu_3 \leq 1 \tag{9}$$

Using the matrix notation of equation (5) we have

$$L = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$

$$b = 1$$

The uncontrolled net does not satisfy the desired constraint since $[0\ 1\ 1]^T$ is not a place invariant of the net. A slack variable μ_s is introduced and the inequality (9) becomes an equality

$$\mu_2 + \mu_3 + \mu_s = 1 \tag{10}$$

The slack variable μ_s denotes the marking of the place p_s which belongs to the controller. Equation (10) represents the desired invariant $x^T = [0\ 1\ 1\ 1]$ which will be forced on the controlled Petri net. The composite change matrix of the controller net is computed with equation (7):

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

The initial marking of the controller place is computed from equation (8):

$$\mu_{s_0} = 1 - L\mu_{p_0} = 1$$

The Petri net graph of the controlled system is shown in figure 1b.

3.2 Maximal Permissiveness

The control method can be shown to be maximally permissive by examining the place invariants of the controlled net. Let X_p be an integer matrix of linearly independent columns representing a basis for the place invariants of the (uncontrolled) process net. Then X_p satisfies the following equation:

$$X_p^T D_p = 0$$

where the columns of X_p are linearly independent, and the number of columns of X_p (and thus the number of invariants) is equal to $n - \text{rank } D_p$ since D_p is an $n \times m$ matrix and X_p forms a basis for the null space of D_p. Note that if rank $D_p = n$ then the uncontrolled plant has no place invariants.

The controller is constructed using equation (7). The transition matrix of the controlled net is then

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} = \begin{bmatrix} D_p \\ -LD_p \end{bmatrix}$$

Note that since the rows of D_c are linear combinations of the rows of D_p, rank D = rank D_p. Thus the number of invariants of the controlled system is equal to $n + n_c -$ rank D_p. All of these invariants are accounted for by the uncontrolled plant invariants and the forced constraint invariants as shown below. First note that

$$\begin{bmatrix} X_p^T & 0 \end{bmatrix} \begin{bmatrix} D_p \\ D_c \end{bmatrix} = X_p^T D_p = 0$$

Thus the invariants of the uncontrolled plant are also invariants of the controlled plant. This is true for any Petri net control scheme that only adds places and arcs in order to control the plant. From the construction of the control law we also know that

$$\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_p \\ D_c \end{bmatrix} = LD_p + D_c = LD_p - LD_p = 0$$

and thus all $n + n_c - \text{rank} D_p$ invariants of the controlled net are given by

$$X_c^T D = \begin{bmatrix} X_p & L \\ 0 & I \end{bmatrix}^T D = 0$$

The rank (and number of columns) of X_c is $n + n_c -$ rankD_p, since X_p is rank $n -$ rankD_p and I is a $n_c \times n_c$ identity matrix.

There are no new or unexpected invariants forced on the system as a result of the control law. The control law is maximally permissive since no action is prohibited which is not a result of the plant structure itself or the constraints forced on the plant.

4 Constraint Transformations

Constraints of the form $L\mu_p \leq b$ are useful for representing a large variety of forbidden state problems and are the same type of constraints discussed by Li and Wonham in [6]. It is possible to transform many system constraints into the form of equation (5) so that the control method presented here can be used to solve the problem. Yamalidou and Kantor have shown in [11] that constraints written as well formed boolean formulas in a "product of sums" form can be transformed into algebraic constraints in the form of equation (5) for safe Petri nets. Some systems may require resource reserve constraints, for example, consider a multiprocessor computer with processor allocation modeled by a Petri net. One constraint on the system might be that two processors must always be available to handle user I/O. The constraint could be written $\mu_i \geq 2$ which is equivalent to $-\mu_i \leq -2$ and is in the form of equation (5). Thus

J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.

"greater than or equal to" constraints are handled easily by the existing control method. The rest of this section details the special cases of equality constraints and constraints that involve elements of the Petri net firing vector.

4.1 Equality Constraints

Equality constraints have the form

$$E\mu_p = k \tag{11}$$

where E and k serve the same functions as L and b in equation (5). Equation (11) defines place invariants on the original process net. This is really a specification for the system and should have been incorporated into the Petri net model. If this invariant is not already part of the Petri net model, it should become one at this point. This is done by modifying the composite change matrix D_p of the Petri net so that equation (2) holds, where $x = E$ in equation (11). The new elements of D_p represent the arcs which should be added to the Petri net so that the place invariants are enforced.

Note that it is possible to use the method from section 3 to force $E\mu_p \leq k$ and $E\mu_p \geq k$ in order to achieve the constraint of equation (11), however this may result in undesired dead-locks in the process net.

4.2 Constraints Involving the Firing Vector

Certain control goals may involve the firing vector of the Petri net as well as the places. For example one might want to insure that two transitions do not fire simultaneously or that a certain transition is never allowed to fire when a certain place holds a token. Algebraic schemes for transforming these kind of constraints into constraints that only involve places can be found in [10]. The method presented here is based on a transformation performed on the Petri net model itself.

Assume that the process net must satisfy the constraint

$$\mu_i + q_j \leq 1 \tag{12}$$

where q_j is the j^{th} element of the firing vector q. This means that transition t_j cannot fire if place p_i is marked and vice versa. In order to bring this constraint to a form which contains elements of the marking vector only, the following transformation is done to the process Petri net. Transition t_j is replaced by two transitions and a place between them, as shown in figure 2.

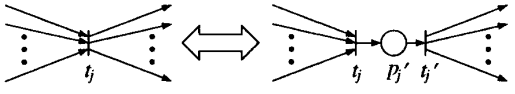


Figure 2: Transformation of a Transition.

The composite change matrix D_p of the process Petri net is increased by one column and one row since the overall number of transitions and places of the Petri net has each been increased by one. This transformation is artificial and will not effect the Petri net model of the process. Its sole purpose is to introduce the place p_j' which records the firing of the transition t_j. After the controller has been computed the process net will be transformed back to its original form.

The marking μ_j' of the place p_j' replaces q_j in the constraint (12), which becomes

$$\mu_i + \mu_j' \leq 1 \tag{13}$$

The constraint now contains only μ's and the controller can be computed as described in section 3. Since the method produces a controller consisting of places and arcs only, no part of the controller is connected directly to the place p_j' of the transformation. After the controller structure is computed, the two transitions and the place of the transformation collapse to the original transition thus restoring the original form of the process net while still maintaining the enforcement of the new constraint. The same transformation is done to all the transitions which appear in the constraints. Those constraints that contain only q's are treated in the same way.

5 Examples

Two examples are used to illustrate the place invariant control method. The first is the well known "cat and mouse" problem [9]. The second is a flexible manufacturing system used by Halloway and Krogh [4].

5.1 The Cat and Mouse Problem

The "cat and mouse" problem, introduced by Wonham and Ramadge [9], is a popular example in the field of discrete event system control. The problem involves a maze of five rooms where a cat and a mouse can circulate. The rooms are connected with doors through which the animals can pass. The problem is to control the doors so that the cat and the mouse can never be in the same room at the same time. The controller should be maximally permissive in the sense that it should grant maximum freedom of movement to both the cat and the mouse. The simple Petri net model of the cat and mouse problem is taken out of Boissel [1] and is shown in figure 3. The upper net concerns the cat while the lower net concerns the mouse. Each net has five places which model the five rooms of the maze. The transitions model the ability of each animal to pass from one room to the other. Further details on this example can be found in [10].

There is one token only in each of the nets since there is one cat and one mouse. The presence of a token in a place indicates that the animal modeled by the token is in the room modeled by the particular place. Initially the cat is in room 2 and the mouse is in room 4.

Transitions c_7 and c_8 are uncontrollable. The control goal is to insure that the cat and mouse are never in the same room simultaneously, while allowing all moves that do not directly violate this constraint, that could lead to the violation of the constraint because of uncontrollability, or that could lead to a dead-lock state. The requirements are translated into the following constraints:

$$\begin{aligned} \mu_0 + \mu_1 + \mu_3 + \mu_5 + \mu_6 + \mu_8 &\leq 1 \\ \mu_0 + \mu_8 &\leq 1 \\ \mu_2 + \mu_7 &\leq 1 \\ \mu_4 + \mu_9 &\leq 1 \end{aligned} \tag{14}$$

J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.

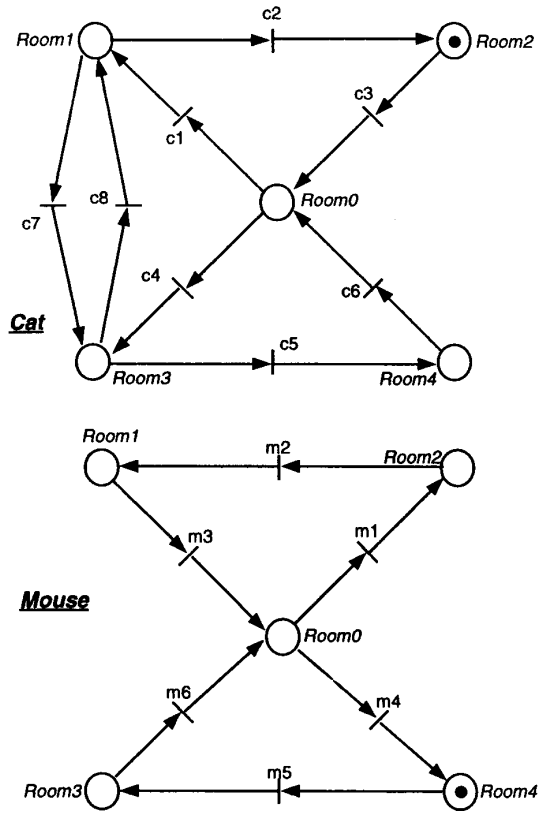


Figure 3: The Petri Net Model of the Cat and Mouse Problem.

The corresponding L and b are

$$L = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Note that even though the second constraint in (14) is contained within the first, the constraint is still necessary to insure that the controller works for all initial conditions. The controller has four places and its composite change matrix and initial conditions are computed as described in section 3:

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

$$\mu_{c0} = \mathrm{b} - \mathrm{L}\mu_{p0} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

5.2 Automated Guided Vehicle Coordination

This example, concerning a flexible manufacturing cell has been used by Holloway and Krogh [4]. It includes three workstations, two part-receiving stations and one completed parts station. There are five automated guided vehicles (AGV's) which can transport material between the stations. The routes of the vehicles cross on the floor of the plant and, consequently, there are zones in which two vehicles could be present at the same time. This is a forbidden situation.

The constraints that the process must satisfy concern the presence of the vehicles in the dangerous zones and are expressed by the following inequalities

$$\begin{aligned} \sum_{i \in Z_1} \mu_i &\leq 1 \\ \sum_{i \in Z_2} \mu_i &\leq 1 \\ \sum_{i \in Z_3} \mu_i &\leq 1 \\ \sum_{i \in Z_4} \mu_i &\leq 1 \end{aligned} \tag{15}$$

where Z_j is the set of indices of places which make up zone j. Slack variables are introduced and the inequalities become equalities

$$\begin{aligned} \sum_{i \in Z_1} \mu_i + \mu_{s_1} &= 1 \\ \sum_{i \in Z_2} \mu_i + \mu_{s_2} &= 1 \\ \sum_{i \in Z_3} \mu_i + \mu_{s_3} &= 1 \\ \sum_{i \in Z_4} \mu_i + \mu_{s_4} &= 1 \end{aligned} \tag{16}$$

The four slack variables define four places for the controller. Each place controls the access of a zone. The composite change matrix of the process is increased by four rows which correspond to the four controller places and constitute the composite change matrix D_c of the controller net. After computing D_c and μ_{c_0} from equations (7) and (8), the appropriate arcs are added to connect the controller places to the appropriate transitions of the Petri net of the process. The controlled Petri net is shown in figure 4. The shaded areas represent the dangerous zones in which the vehicles' trajectories cross. The vehicles and the parts are modeled by tokens (see [4]). Arcs made of dashed lines are parts of the controller, not the original process net. The marking of the Petri net corresponds to the current state of the system.

6 Conclusions

This paper has presented a particularly simple method for constructing feedback controllers for untimed Petri nets. The method is based on the idea that specifications representing desired plant behaviors can be enforced by making them invariants of the controlled net. In this paper, therefore, a technique was derived which used place invariants representative of logical design specifications. The resulting controller consists only of places and arcs and its size is proportional to the number of constraints.

The significance of this particular approach to Petri net controller design is that the desired control net can be computed very efficiently by a single matrix multiplication (and some simple vector arithmetic to determine

J. Moody, K. Yamalidou, M. D. Lemmon and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants," P roc o f t he 3 3rd I EEE C o nference o n D ecision a nd C ontro l , pp 3104-3109, Lake Buena Vista, FL, Dec 14-16, 1994.

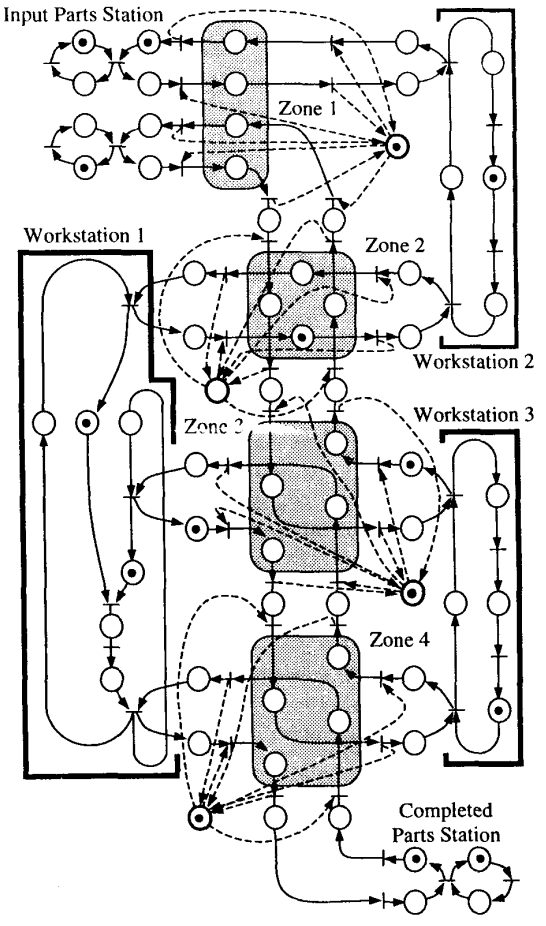


Figure 4: The Controlled AGV Net.

the initial state of the controller). The resulting controlled system will generally not be optimal in terms of minimizing the number of its places in the controller net. However, the approach yields controllers whose size grows in a polynomial manner with the number of specifications and due to the ease of computation can represent a good initial point in subsequent controller-size optimization. Consequently, the proposed approach appears to possess significant potential for helping in the design of feedback controllers for a relatively large class of Petri nets.

The type of constraints that can be written in the form of equation (5) cover a wide variety of forbidden state problems. However constraints in this form may be too cumbersome or even incapable of representing certain, more arbitrary, forbidden state problems. Certain DES control goals such as formal language realization can not be achieved directly by the method presented here and require the addition of some supplemental control mechanism such as a memory. This is a topic for future research.

There are several other areas in which the control method is open for further research. The method should be expanded to deal with timed Petri nets as these networks are being used more and more often due to their added modeling power. Another important research goal is to deal with uncontrollability (and unobservability) in a systematic way, though many times the uncontrollability problem can be dealt with in an *ad hoc* fashion as in section 5.1. A systematic method for dealing with controllability and unobservability could be based on transforming the set of constraints into a set that accounts for uncontrollable or unobservable transitions. This transformation would be similar to the "supremal controllable sublanguage" used by Ramadge and Wonham in their work on discrete event system control [9, 7] and would require the prevention of dead-lock states as well as those states expressly prohibited by the design constraints.

References

[1] Boissel O., "Optimal Feedback Control Design for Discrete-Event Process Systems Using Simulated Annealing", M.S. Thesis, University of Notre Dame, Notre Dame, Indiana, 1988.

[2] Giua, A. and DiCesare, F., "Blocking and Controllability of Petri Nets in Supervisory Control", *IEEE Trans. Automat. Contr.*, 39(4), pp. 818-823, April 1994.

[3] Holloway, L.E., and Krogh, B.H., "Controlled Petri Nets: A Tutorial Survey", *Lecture Notes in Computer Science*, v. 199, G. Cohen and J-P Quadrat eds., pp. 158-168, Eleventh International Conference on Analysis and Control, Discrete Event Systems, June 1994.

[4] Holloway, L.E., and Krogh, B.H., "Synthesis of Feedback Logic for a Class of Controlled Petri Nets", *IEEE Trans. Automat. Contr.*, 35(5), pp. 514-523, 1990.

[5] Lautenbach O., "Linear Algebraic Techniques for Place/Transition Nets", *Lecture Notes in Computer Science*, vol 254, pp 142-167, 1987.

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

[7] Ramadge, P. J. G., Wonham, W. M., "The Control of Discrete Event Systems," *Proceedings of the IEEE*, vol 77, No. 1, pp 81–97, January 1989.

[8] Reisig W., "Petri Nets", Springer-Verlag, 1985.

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

[10] Yamalidou, E., Moody, J.O., Lemmon, M., Antsaklis, P., "Feedback Control of Petri Nets Based on Place Invariants," Technical report ISIS-94-002.2 of the ISIS Group at the University of Notre Dame, Notre Dame, IN, May 1994.

[11] Yamalidou, E. and Kantor, J.C., "Modeling and Optimal Control of Discrete-Event Chemical Processes Using Petri Nets", *Comp. Chem. Engng.*, 15(7), pp. 503-519, 1991.