

SYNTHESIS OF DEADLOCK PREVENTION SUPERVISORS USING PETRI NETS*

Marian V. Iordache^{1,2}, John O. Moody³, Panos J. Antsaklis²

Abstract— Given an arbitrary Petri net structure, which may have uncontrollable and unobservable transitions, the deadlock prevention procedure presented here determines a set of linear inequalities on the Petri net markings. When the Petri net is supervised so that its markings satisfy these inequalities, the supervised net is proved to be deadlock-free for all initial markings that satisfy the supervision constraints. Deadlock-freedom implies that there will always be at least one transition that is enabled in the closed loop (supervised) system. The method is not guaranteed to insure liveness, as it can be applied to systems that cannot be made live under any circumstances. However, it is shown that when the method does insure liveness, it is at least as permissive as any other liveness-insuring supervisor. Moreover, it is shown that the method is not restrictive even for Petri nets in which not all transitions can be made live. The procedure allows automated synthesis of the supervisors.

I. INTRODUCTION

We consider a procedure for the automatic generation of deadlock prevention supervisors for arbitrary Petri net structures. These supervisors are specified independently of the initial marking, prevent deadlock, and are permissive. Deadlock prevention means that the closed loop plant/supervisor system is deadlock-free, that is, we avoid all deadlock states and all states from which deadlock is unavoidably reached. The results presented in this paper are novel and to the authors knowledge are superior to related results in the literature.

The deadlock prevention method presented here uses Petri net models for the plant and results in a Petri net model of the supervisor, providing a unified formalism for representing the closed-loop system. The method presents the conditions necessary to insure deadlock freedom as a set of linear integer inequalities. Such formulation is important because it can be used directly in optimization problems, e.g., determining the minimum number of resources a system requires using a linear integer program. The method is flexible enough to incorporate desired constraint specifications on the markings of the plant. This method is appropriate for use on nets that may not be structurally live, i.e., nonrepetitive systems for which liveness cannot be enforced under any circumstances. When the procedure is applied to repetitive systems, complete system liveness may be the result (some sufficient conditions may be found in [6]). We show that the resulting supervisor is at least

as permissive as any liveness-enforcing supervisor, i.e., no liveness-insuring supervisor will ever allow a transition to fire that our procedure would prevent from firing. Thus, when the procedure enforces liveness, it can be said to be a “maximally permissive” liveness supervisor. The procedure presented here can be computationally expensive, however, all computations are performed off-line. This differentiates our technique from deadlock *avoidance* strategies that perform potentially expensive computations while the system is in operation. A supervisor resulting from our deadlock *prevention* method requires very little in terms of computational resources at run time. The method is an iterative approach that removes new potential deadlock situations at every iteration. When (and if) the procedure terminates, the control designer is presented with either a supervised net that is guaranteed to be deadlock-free or, in the case of fully controllable and observable systems, with an indication that the plant cannot be made deadlock-free under any circumstances.

The method we use generates linear marking inequalities that define the supervisor. The supervisor is built by using the place invariant based methodology of [8], [14], [4]. It is well known that deadlock in Petri nets is related to *siphons* (e.g. [12]). As in other previous methodologies, e.g. [1], [3], [7], we use control places to prevent the total marking in the siphons from becoming zero. In [3] it has been noticed that such siphon control is not enough to guarantee deadlock prevention, since new siphons may appear by adding control places. This problem has been solved in [3] for a subclass of bounded and conservative Petri nets by using more restrictive control policies, and liveness enforcement has thus been achieved. In [7] successive control of the siphons is used, until no new siphons appear. This is also one of the ideas of our procedure. One of the problems which appear by successively controlling the siphons in an ordinary Petri net is that the Petri net can stop being ordinary. Controlling siphons in a Petri net which is not ordinary is harder. A related result is given in [2], but we cannot use it as we desire our method to be as permissive as possible. Instead we transform the Petri net at the different stages of the procedure back into ordinary Petri nets, by adapting a technique from [7]. In order to have a method effective for nonrepetitive Petri nets, we define certain repetitive subnets of a Petri net as active subnets. Based on this idea, we then define a subclass of siphons, called active siphons, and prove new results which are fundamental for our method. These developments can be found in section II. The procedure that leads to deadlock free Petri nets is described in section III and is illustrated via an example in section IV. We give

* Submitted as regular paper. The partial financial support of the National Science Foundation (ECS95-31485) and the Army Research Office (DAAG55-98-1-0199) is gratefully acknowledged.

¹Corresponding author

²Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 (e-mail: iordache.1, antsaklis.1@nd.edu)

³Lockheed Martin Federal Systems, 1801 State Rt.17C, MD 0210, Owego NY 13827-3998 (e-mail: john.moody@lmco.com)

the theoretical results in section V. Some important proofs are included in the appendix.

Finally, note that when the Petri nets are bounded and the initial marking is fixed, it is possible to transform the problem from the Petri net framework to finite automata, and so to solve the problem by using finite automata methodologies, for instance supervisory control techniques. However, the approach presented here makes no assumptions about the Petri net structure: the Petri net may be unbounded, its arc weights may not be one, it may be non-repetitive (that is supervision for liveness enforcement is impossible for all initial markings), and it may have uncontrollable and unobservable transitions. Furthermore the usage of Petri nets in deadlock prevention may be preferable because deadlock often occurs in systems with concurrency, which are best described by Petri nets.

II. PRELIMINARY RESULTS

In this section we first define a class of subnets of a Petri net, and then a subclass of siphons. Then in Proposition 2.1 we prove an important deadlock property, which is fundamental for our deadlock prevention approach. We consider Petri net structures of the form $\mathcal{N} = (P, T, F, W)$, where P is the set of places, T the set of transitions, F the set of transition arcs and W the weight function. Let D be the incidence matrix of the Petri net and consider the usual Petri net notations [11].

In this paper we focus on enforcing (rather than verifying) deadlock-freedom in Petri nets. Thus we need to introduce the following conventions. We say that a **Petri net can be made deadlock-free/live** if there is a supervisory policy and an initial marking μ_0 such that the supervised Petri net is deadlock-free/live. In a Petri net a **transition t can be made live** if there is a supervisory policy and an initial marking μ_0 such that t is live in the supervised Petri net.

A Petri net \mathcal{N} is **ordinary** if $\forall f \in F : W(f) = 1$. We will need to refer to slightly more general Petri nets in which only the arcs from places to transitions have weights equal to one. We call such (partially ordinary) Petri nets *PT-ordinary*. The deadlock prevention procedure of section III applies to arbitrary Petri net structures, not necessarily PT-ordinary, however it includes a transformation of general Petri nets to PT-ordinary (section III-B).

Definition 2.1 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. We call \mathcal{N} **PT-ordinary** if $\forall p \in P, \forall t \in T$, if $(p, t) \in F$ then $W(p, t) = 1$.*

A Petri net is said to be **(partially) repetitive** [11] if a marking μ_0 and an infinite firing sequence σ from μ_0 exist, such that every (some) transition occurs infinitely often in σ . A Petri net is (partially) repetitive if and only if a vector x of positive (nonnegative) integers exists, such that $D \cdot x \geq 0$, $x \neq 0$ [11]. From this result we have derived the following corollary, proved in [6].

Corollary 2.1 *Consider a Petri net $\mathcal{N} = (P, T, F, W)$ which is not repetitive, and let D be the incidence matrix. Then at least one transition exists such that for any given*

initial marking it cannot fire infinitely often. Let T_D be the set of all such transitions. There is a nonnegative integer vector x such that $Dx \geq 0$, $x(i) \neq 0 \forall t_i \in T \setminus T_D$ and $x(i) = 0 \forall t_i \in T_D$.

In what follows we define a class of subnets of a Petri net, which we call *active subnets*. An active subnet can be made live by supervision for appropriate initial markings. Note that the notation $\bullet S$ ($S \bullet$) refers to the preset (postset) of S .

Definition 2.2 *Let $\mathcal{N} = (P, T, F, W)$ be a Petri net, D the incidence matrix and $T_D \subseteq T$ the set defined in Corollary 2.1. $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ is an **active subnet** of \mathcal{N} if $P^A = T^A \bullet$, $F^A = F \cap \{(T^A \times P^A) \cup (P^A \times T^A)\}$, W^A is the restriction of W to F^A and T^A is the set of transitions with nonzero entry in some nonnegative vector x which satisfies $Dx \geq 0$. The **maximal active subnet** of \mathcal{N} is the active subnet $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ such that $T^A = T \setminus T_D$.*

Given a Petri net, a simple algorithm based on linear programming can be employed to compute the maximal active subnet, as we show in [6]. A **siphon** is a set of places S such that $\bullet S \subseteq S \bullet$. Next we introduce a particular type of siphon.

Definition 2.3 *Given an active subnet \mathcal{N}^A of a Petri net \mathcal{N} , a siphon of \mathcal{N} is said to be an **active siphon** with respect to \mathcal{N}^A if it is, or includes, a siphon of \mathcal{N}^A . An active siphon is **minimal** if it does not include another active siphon with respect to the same active subnet.*

The next lemma is necessary for the proof of Proposition 2.1. We prove it in [6].

Lemma 2.1 *Let $\mathcal{N}^A = (P^A, T^A, F^A, W^A)$ be an active subnet of \mathcal{N} . Given a marking μ of \mathcal{N} , μ^A its restriction to \mathcal{N}^A and $t \in T^A$, μ^A enables t in \mathcal{N}^A if and only if μ enables t in \mathcal{N} .*

It is known that a deadlocked ordinary Petri net has an empty siphon [12]. Unfortunately this result is a little too general for dealing with deadlock prevention in nonrepetitive Petri nets. The following new result identifies the specifically problematic siphons.

Proposition 2.1 *Let \mathcal{N}^A be an arbitrary, nonempty, active subnet of a PT-ordinary Petri net \mathcal{N} . If μ is a deadlock marking of \mathcal{N} , then there is at least one empty minimal active siphon with respect to \mathcal{N}^A .*

Proof: Since μ is a deadlock marking and $\mathcal{N} = (P, T, F, W)$ is PT-ordinary, $\forall t \in T \exists p \in \bullet t: \mu(p) = 0$. The active subnet is built in such a way that if the marking μ restricted to the active subnet enables a transition t , then μ enables t in the total net (Lemma 2.1.) Therefore, because the total net (\mathcal{N}, μ) is in deadlock, the active subnet is deadlocked, and so there is an empty minimal siphon s of the active subnet. Consider s in the total net. If s is a siphon of the total net, then s is also a minimal active siphon; therefore the net has a minimal active siphon which is empty. If s is not a siphon of the total net: $\bullet s \setminus T^A \neq \emptyset$.

Let S be the set recursively constructed as follows: $S_0 = s$, $S_i = S_{i-1} \cup \{p \in \bullet(\bullet S_{i-1} \setminus S_{i-1} \bullet) : \mu(p) = 0\}$, where μ is the (deadlock) marking of the net. In other words S is a completion of s with places with null marking such that S is a siphon. By construction S is an active siphon and is empty for the marking μ . Hence an empty minimal active siphon exists. \blacksquare

The significance of Proposition 2.1 is that it provides a way to do deadlock prevention, since deadlock is impossible when all active siphons with respect to a nonempty active subnet cannot become empty.

III. THE DEADLOCK PREVENTION APPROACH

A. Introduction to the Method

Given a Petri net \mathcal{N}_0 , the deadlock prevention procedure generates a sequence of PT-ordinary Petri nets, $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$, increasingly improved with respect to deadlock prevention. \mathcal{N}_1 is \mathcal{N}_0 transformed into a PT-ordinary net. The other Petri nets are obtained as follows: at each iteration i the new minimal active siphons of \mathcal{N}_i are *controlled*, and then, if needed, transitions are split; the resulting PT-ordinary net is \mathcal{N}_{i+1} . The active siphons (see Definition 2.3) of each \mathcal{N}_i are taken with respect to an active subnet \mathcal{N}_i^A computed for every iteration i . To control a siphon, a linear marking inequality is enforced. Let $L_i \mu \geq b_i$ be the total set of constraints enforced in \mathcal{N}_i . Because \mathcal{N}_k is the last Petri net in the sequence, it has no uncontrolled active siphons. Therefore \mathcal{N}_k is deadlock free for all initial markings which satisfy $L_k \mu \geq b_k$. Finally, the constraints defined by (L_k, b_k) can be easily translated to constraints in terms of the markings of \mathcal{N}_0 ; these constraints define the supervisor for deadlock prevention in \mathcal{N}_0 .

The procedure supports (linear inequality) **initial constraints**, that is additional desired specifications may be incorporated in the method. The usage of initial constraints $L_I \mu \geq b_I$ may result into less complex supervisors, it may enhance convergence and it also guarantees that the procedure will not generate constraints which require $L_I \mu \not\geq b_I$.

B. Transforming Petri Nets to PT-ordinary Petri Nets

We use transition splits to transform Petri nets to PT-ordinary Petri nets. The transformation is a modification of a similar operation in [7]. Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. Transitions $t_j \in T$ such that $W(p, t_j) > 1$ for some $p \in \bullet t_j$ are **split**. Given t_j , let $m = \max\{W(p, t_j) : p \in \bullet t_j\}$. When t_j is split, t_j is enhanced with $m - 1$ new transitions $t_{j,1}, t_{j,2}, \dots, t_{j,m-1}$ and $m - 1$ new places $p_{j,1}, p_{j,2}, \dots, p_{j,m-1}$. Let $\mathcal{N}' = (P', T', F', W')$ be the new Petri net obtained by splitting t_j . The connections are as follows, where the preset/postset operator is denoted by \bullet for evaluations in \mathcal{N} , and by \bullet' in \mathcal{N}' .

- (i) $\bullet' p_{j,i} = t_{j,i}$ and $t_{j,i} \bullet' = p_{j,i}$ for $i = 1 \dots m - 1$, $p_{j,i} \bullet' = t_{j,i-1}$ for $i = 2 \dots m - 1$ and $p_{j,1} \bullet' = t_j$.
- (ii) $\bullet' t_{j,i} = \{p \in \bullet t_j : W(p, t_j) > i\} \cup Y$, for $i = 1 \dots m - 1$, where $Y = \emptyset$ for $i = m - 1$ and $Y = \{p_{j,i+1}\}$ otherwise.
- (iii) $\bullet' t_j = \bullet t_j \cup \{p_{j,1}\}$ and $t_j \bullet' = t_j \bullet$.

(iv) $\forall p \in \bullet' t_{j,i} : W'(p, t_{j,i}) = 1$ and $W'(t_{j,i}, p_{j,i}) = 1$, for $i = 1 \dots m - 1$.

(v) $\forall p \in \bullet' t_j : W'(p, t_j) = 1$ and $\forall p \in t_j \bullet' : W'(t_j, p) = W(t_j, p)$.

Note that the connections of t_j in \mathcal{N}' are the same as in \mathcal{N} , except for an additional transition arc and for the weights of the input arcs. Firing t_j in \mathcal{N} corresponds to firing the sequence $t_{j,m} \dots t_{j,1}, t_j$ in \mathcal{N}' . An example of transition split is in section IV.

C. Enforcing Linear Marking Constraints

A linear constraint on the marking vector has the form $L\mu \geq b$, where L is matrix and b is vector. Enforcing such constraints is done according to the supervision based on place invariants in [8], [14]. However this requires admissible constraints. A constraint is **admissible** [8], if the supervisor enforcing the constraint does not inhibit an enabled uncontrollable transition and does not observe an unobservable transition. A constraint $L\mu \geq b$ is admissible if the following conditions of [8] are satisfied: $LD_{uc} \geq 0$ and $LD_{uo} = 0$, where the columns of D_{uc} and D_{uo} are the columns of the incidence matrix D that correspond to uncontrollable (D_{uc}) and unobservable (D_{uo}) transitions.

A siphon S is said to be **controlled** [1], [2] if

$$\sum_{p \in S} \mu(p) \geq 1 \quad (1)$$

is true for all reachable markings μ . This is a linear constraint, and enforcing it via the methods in [8], [14] is equivalent to the approach used for siphon control in [1], [3], [7]; note that the latter references consider controllable and observable transitions only. In the case of uncontrollable and unobservable transitions we need to check that this constraint is admissible, and if not to transform it to an admissible constraint. Indeed, in order that the constraints $L\mu \geq b$ produced by the deadlock prevention procedure define a valid supervisor of the target Petri net, we need $L\mu \geq b$ to be admissible. Recall that the procedure adds constraints to the intermediary Petri nets \mathcal{N}_i , and when it terminates, the final set of constraints is written in terms of the target net \mathcal{N}_0 . Thus we are interested in the case when the constraint (1), when written in terms \mathcal{N}_0 , is admissible. Since this may not always be the case, we provide in [6] an algorithm to transform (1) to an admissible constraint

$$\sum_{p \in S} \alpha_p \mu(p) \geq 1 \quad (2)$$

such that $\alpha_p \in \mathbb{N}$ and at least two coefficients α_p are nonzero. Enforcing (2) requires an additional place, which is called **control place**. The control place C of a siphon S introduces the place invariant described by

$$\mu(C) = \sum_{p \in S} \alpha_p \mu(p) - 1 \quad (3)$$

We are interested in working with PT-ordinary Petri nets, so for all $t \in C \bullet$ such that $W(C, t) > 1$, t is split. The

inequality (2) is still true after the split, but the place invariant is changed to include the markings of the new places resulting through the split and (3) is changed accordingly. Consider an enforced inequality $l^T \mu \geq b$ or an invariant $l^T \mu = b$, where $l \in \mathbb{N}^{n \times 1}$, $b \in \mathbb{N}$ and n is the number of places. If a transition t_i is split, using the notations of section III-B, the inequality or invariant is modified as follows:

$$\sum_p l_p \mu(p) \rightarrow \sum_p l_p \mu(p) + \sum_{p, m_p > 1} \left(l_p \sum_{j=1}^{m_p-1} j \mu(p_i, m_p-j) \right) \quad (4)$$

where $m_p = W(p, t_i)$ if $p \in \bullet t_i$ and else $m_p = 0$. ($W(p, t_i)$ is evaluated before splitting t_i .)

In an intermediary Petri net \mathcal{N}_i , the marking of the control places μ_c can be expressed in terms of the marking of the other places μ_p by

$$\mu_c = L_i \mu_p - b_i \quad (5)$$

The matrices L_i and b_i are recursively obtained as follows: if a control place C has been added in iteration i with regard to a siphon S , replace in (3) the markings of all control places $C' \in S$ added in the previous iterations with their expressions available from L_{i-1} and b_{i-1} . Thus the new form of (3) is:

$$\mu(C) = l^T \mu_p - b \quad (6)$$

D. The Deadlock Prevention Procedure

Input: The target Petri net $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ and a possibly empty set of initial constraints (L_I, b_I) .

Output: Two sets of constraints (L, b) and (L_0, b_0) . (Deadlock is prevented for all initial markings μ_0 such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, when (\mathcal{N}_0, μ_0) is supervised according to $L\mu \geq b$, where the constraints $L\mu \geq b$ are by construction admissible.)

Procedure:

A. (L_0, b_0) is initialized to (L_I, b_I) and (L, b) to be empty. \mathcal{N}_0 is transformed into a PT-ordinary net; the new Petri net is \mathcal{N}_1 and (L_0, b_0) is updated accordingly (see relation (4)). Let $i = 1$. If not previously defined, let $X = \emptyset$.

B. The largest active subnets of \mathcal{N}_0 and \mathcal{N}_1 which do not contain the transitions of X are computed. Let them be \mathcal{N}_0^A and \mathcal{N}_1^A . If \mathcal{N}_0^A is empty, the procedure terminates: deadlock cannot be prevented in \mathcal{N}_0 under any circumstances.

C. **For** $i \geq 1$ **do** (the initial Petri net of the iteration i is denoted $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$ and the active subnet \mathcal{N}_i^A .)

C.1 If no new uncontrolled minimal active siphon is found, the next step is step D below. (The active siphons are taken with respect to the current active subnet \mathcal{N}_i^A . A siphon S is *uncontrolled* if (1) is not implied by the current $L\mu \geq b$ and $L_0\mu \geq b_0$)

C.2 **For** every new uncontrolled minimal active siphon S **do**

C.2.a Let χ be the constraint (1). It is checked whether enforcing χ produces a control place C such that $C \bullet \subseteq \bullet S$. If so, S does not need control, C is not added to \mathcal{N}_i , and the next step is step C.2.d below.

C.2.b If χ is an inadmissible constraint, χ is transformed so that it is an admissible constraint of the form (2). If this is not possible, $X \rightarrow X \cup S \bullet$ and the *for* loop continues with the next active siphon.

C.2.c The constraint χ is enforced using the invariant based supervision [8], [14].

C.2.d Let $l^T \mu \geq c$ be the constraint χ written in the form (6). It is checked that the system $l^T \mu \geq c$, $L\mu \geq b$, $L_0\mu \geq b_0$ is feasible. If the system is infeasible, $X \rightarrow X \cup S \bullet$. Else, $l^T \mu \geq c$ is added to (L_0, b_0) if the previous step was C.2.a, or to (L, b) if the previous step was C.2.c.

C.3 If the Petri net is no longer PT-ordinary, the transitions which do not comply with this requirement are *split* (section III-B.) The matrices L and L_0 are enhanced with null columns, each column corresponding to one new place resulting from the transition split.

C.4 The active subnet is updated as the largest active subnet which does not contain the transitions in X .

C.5 Let T^A be the set of transitions of the active subnet. If the active subnet is empty ($T^A = \emptyset$), the procedure cannot generate a deadlock prevention supervisor and so it terminates. Else if an infeasibility occurred at a step C.2.d of the current iteration, $X \rightarrow T_0 \setminus T^A$ and the procedure is restarted at the step A with this value of X .

C.6 The final nets of the iteration i are denoted by \mathcal{N}_{i+1} and \mathcal{N}_{i+1}^A . The next step is C.1 for $i \rightarrow i + 1$.

D. The constraints (L, b) and (L_0, b_0) are modified to be written only in terms of the marking of the target net \mathcal{N}_0 . This is done by removing the columns of L and L_0 corresponding to places not in \mathcal{N}_0 .

E. Redundant constraints of (L, b) and (L_0, b_0) are removed using integer programming techniques.

IV. A FLEXIBLE MANUFACTURING EXAMPLE

Consider the target Petri net structure of figure 1(a), where the marking of the places p_3 , p_6 and p_7 corresponds to available resources. The transition t_{10} is uncontrollable. Next we show the operations performed by the procedure, in view of section III-D.

In the first iteration, the Petri net structure $\mathcal{N}_1 = (P_1, T_1, F_1, W_1)$ is that of figure 1(b), but without the control places C_1, \dots, C_4 and their transition arcs. The place $p_{2,1}$ and the transition $t_{2,1}$ appeared by splitting t_1 . The maximal active subnet has the transitions in $T_1 \setminus \{t_9, t_{10}\}$. There are two minimal active siphons: $\{p_1, p_6\}$ and $\{p_4, p_7, p_8\}$. They are controlled with two new control places: C_1 and C_2 respectively, where the constraint of C_2 is transformed to be admissible.

In the second iteration, the maximal active subnet still has the transitions $T_1 \setminus \{t_9, t_{10}\}$ and the only new minimal active siphon is $\{C_2, p_8\}$. There is no admissible constraint of the form (2) for the control of $\{C_2, p_8\}$. ($\mu(C_2) \geq 1$ is not of the form (2), as (2) requires at least two nonzero

coefficients α_p .) Therefore X , the set of transitions which should not appear in the active subnets of the following iterations, is set to $X = \{t_5, t_7\}$.

In the third iteration and the remaining iterations the active subnet has the set of transitions $T_1 \setminus \{t_5, t_7, t_9, t_{10}\}$. The only new minimal active siphon is $S = \{C_2, p_8, p_3, p_5\}$. The control place which results is C_3 , enforcing the constraint (1), which is admissible in this case.

In the fourth iteration the only new minimal active siphon is $\{p_1, C_1, p_5, C_3\}$, and so the control place C_4 is added.

In the fifth iteration the only new minimal active siphon is $S = \{C_4, p_{1,1}, p_5\}$. Since the control place which would control this siphon satisfies $C \bullet \subseteq \bullet S$, no control place is added, and the constraint (1) is included in (L_0, b_0) .

The procedure terminates at the sixth iteration, as there is no new minimal active siphon. The constraints at the step D of the procedure are:

$$\mu(p_1) + \mu(p_6) \geq 1 \quad (7)$$

$$\mu(p_4) + \mu(p_7) \geq 1 \quad (8)$$

$$\mu(p_3) + \mu(p_4) + \mu(p_5) + \mu(p_7) + \mu(p_8) \geq 2 \quad (9)$$

$$2\mu(p_1) + \mu(p_3) + \mu(p_4) + 2\mu(p_5) + \mu(p_6) + \mu(p_7) + \mu(p_8) \geq 4 \quad (10)$$

$$2\mu(p_1) + \mu(p_3) + \mu(p_4) + 3\mu(p_5) + \mu(p_6) + \mu(p_7) + \mu(p_8) \geq 5 \quad (11)$$

The inequalities (7-10) are included in $L\mu \geq b$, and correspond to $C_1 \dots C_4$ in this order, while the inequality (11) is written as $L_0\mu \geq b_0$. The inequality (10) is redundant, and so it can be omitted. The Petri net supervised for deadlock freedom is obtained by enforcing the constraints (L, b) on the target net (figure 1(c)).

V. MAIN RESULTS

In this section we present the theoretical results that support the approach described in section III. We prove that our new deadlock prevention method produces supervisors which prevent deadlock, we prove that the supervisors are not restrictive and we introduce a modification of our approach for guaranteed termination.

Definition 5.1 A marking μ of an intermediary Petri net \mathcal{N}_i is said to be **forbidden** if its restrictions to the control places (μ_c) and the rest of the places (μ_p) do not satisfy equation (5). A marking μ of \mathcal{N}_i is **valid** if it is not forbidden and if $\mu(p) \neq 0$ only if p is a place of \mathcal{N}_0 or a control place. The markings μ_i of \mathcal{N}_i and μ_j of \mathcal{N}_j are **equivalent** if both are valid and $\mu_i(p) = \mu_j(p)$ for all places p common to \mathcal{N}_i and \mathcal{N}_j .

In view of the following proofs we need to specify some notations. When a transition t_i of \mathcal{N} is split in $t_{i,1}, \dots, t_{i,m_i}$, thus a new net \mathcal{N}' resulting, firing the sequence $t_{i,m_i}, \dots, t_{i,1}, t_i$ has the same effect as firing t_i in \mathcal{N} . In our procedure a transition t_i may be split in some iteration, then some $t_{i,k}$ (where $t_{i,k}$ resulted by splitting t_i) can be split in a subsequent iteration, and so on. We denote by $\sigma_{0,j}(t)$ an arbitrary transition sequence of \mathcal{N}_j such that (a) $\sigma_{0,j}(t)$ enumerates the transitions (including t itself) in which t of \mathcal{N}_0 is successively split until (and including) the

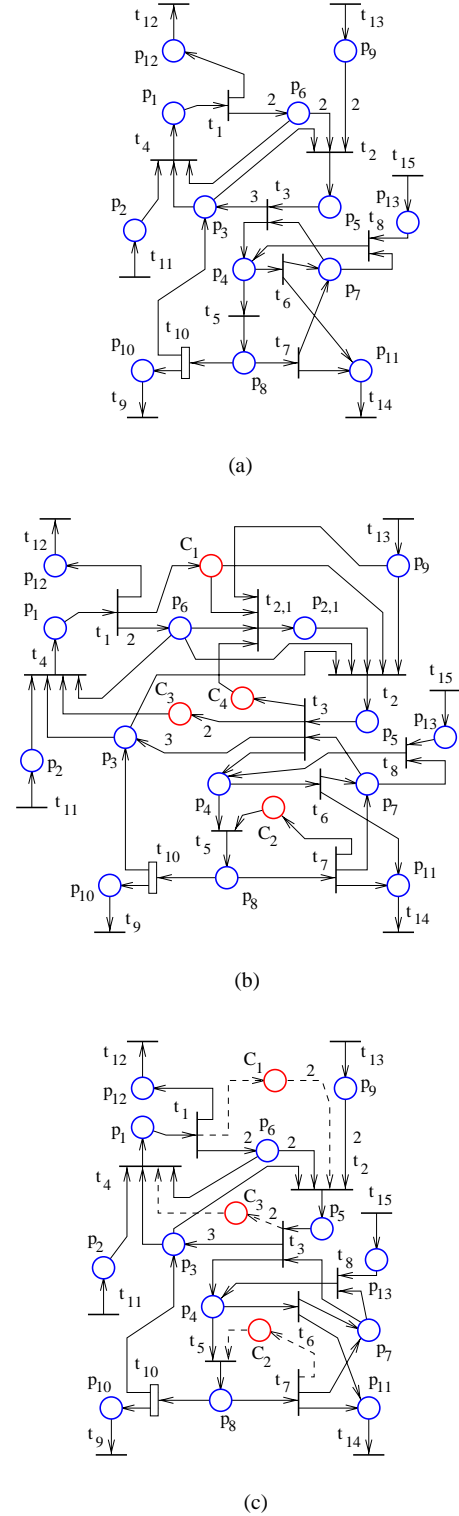


Fig. 1. (a) Target Petri net, (b) the Petri net after five iterations and (c) the supervised Petri net.

iteration $j - 1$, and (b) markings μ of \mathcal{N}_j exist such that $\mu(p) \neq 0$ only if p is a place of \mathcal{N}_0 or a control place, and μ enables $\sigma_{0,j}(t)$. In this way firing the sequence $\sigma_{0,j}(t)$ in \mathcal{N}_j corresponds to firing t in \mathcal{N}_0 . If t is not split, we let $\sigma_{0,j}(t) = t$. The notation $\sigma_{i,j}(t)$ for $i < j$ and t in \mathcal{N}_i , is similarly defined by taking \mathcal{N}_i instead of \mathcal{N}_0 . Also, if $\sigma = t_1 t_2 t_3 \dots$, we let $\sigma_{i,j}(\sigma) = \sigma_{i,j}(t_1) \sigma_{i,j}(t_2) \sigma_{i,j}(t_3) \dots$. For all $i \geq 0$ we use the notation $\mathcal{N}_i = (P_i, T_i, F_i, W_i)$.

Theorem 5.1 *Assume that the procedure terminates. Let \mathcal{N}_0 be the original Petri net and \mathcal{N}_k the net produced by the last iteration. Let (L, b) and (L_0, b_0) denote the two sets of constraints generated by the procedure. If \mathcal{N}_k^A is nonempty, then the original net \mathcal{N}_0 , in closed loop with the supervisor enforcing $L\mu \geq b$, is deadlock-free for all initial markings μ_0 of \mathcal{N}_0 such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$.*

Proof: Let μ_0 be an arbitrary initial marking of \mathcal{N}_0 satisfying the constraints and let $\mu_{0,k}$ be the equivalent initial marking in \mathcal{N}_k , that is (a) $\mu_0(p) = \mu_{0,k}(p) \forall p \in P_0$, (b) for each control place C , $\mu_{0,k}(C)$ satisfies (3), and (c) $\mu_{0,k}(p) = 0 \forall p \in P_k \setminus (P_0 \cup \mathcal{C})$, where \mathcal{C} is the set of control places in \mathcal{N}_k . By construction, all active siphons of $(\mathcal{N}_k, \mu_{0,k})$ are controlled. Indeed, the split transition operation has been chosen such that if a transition t is split, a controlled minimal active siphon remains controlled and minimal active siphon. Therefore controlling only the new active siphons is enough to guarantee that at the end all minimal active siphons of \mathcal{N}_k (with respect to \mathcal{N}_k^A) are controlled. Hence, in view of Proposition 2.1, $(\mathcal{N}_k, \mu_{0,k})$ is deadlock-free.

We prove by contradiction that it is impossible that (\mathcal{N}_0, μ_0) in closed loop with the supervisor defined by $L\mu \geq b$ reaches a marking μ such that all possible firings in \mathcal{N}_0 lead either to deadlock markings or to markings which do not comply with the enforced constraints, $L\mu \geq b$. Assume the contrary, that such a marking μ can be reached. Let μ_k be the equivalent marking of μ in \mathcal{N}_k . Because (\mathcal{N}_k, μ_k) is deadlock free, μ_k enables an infinite transition sequence σ in \mathcal{N}_k . Let T_R be the set of transitions created by split transition operations. Enforcing (2) on a siphon S yields $C \notin T_R \bullet$ by the way we construct (2) [6]; we also prove in [6] that enforcing (1) yields $C \notin T_R \bullet$. Therefore firing any $t \in T_R$ always reduces the marking of some places in $P_0 \cup \mathcal{C}$ and only firing $t \in T_0$ (note that $T_0 = T_k \setminus T_R$) may increase the marking of some places in $P_0 \cup \mathcal{C}$. Because the total marking of $P_0 \cup \mathcal{C}$ is finite, σ must include transitions $t \in T_0$. Let t_1 be the first transition in T_0 that appears in σ . Since all transition of σ before t_1 are in T_R , firing them only decrease markings of $P_0 \cup \mathcal{C}$, and t_1 cannot fire unless all other transitions of $\sigma_{0,k}(t_1)$ fired before (as μ_k is valid), it follows that $\sigma_{0,k}(t_1)$ is enabled by μ_k . But this implies that t_1 also is enabled by μ in \mathcal{N}_0 supervised with $L\mu \geq b$, which is a contradiction. ■

The assumptions of Theorem 5.1 are that the procedure terminates and that the final active subnet \mathcal{N}_k^A is not empty. The next result considers the case when the procedure terminates and \mathcal{N}_k^A is empty.

Proposition 5.1 *Deadlock cannot be prevented under any*

circumstances if \mathcal{N}_0^A is empty, or if \mathcal{N}_0 has no uncontrollable and unobservable transitions and \mathcal{N}_k^A is empty.

Proof: The first part is a consequence of Corollary 2.1. The second part is a consequence of Lemma 5.1, as we show in what follows. Using the same idea as in the proof of Theorem 5.2 (see appendix), any transition t which can be made live for some marking satisfying the initial constraints has the property that for all iterations i , neither t , nor one of the transitions in which t may be split, are in the postset of an active siphon S of \mathcal{N}_i which must be empty. (A siphon S must be empty if the inequality (1) conflicts with the initial constraints.) Since \mathcal{N}_k^A is empty, in view of the steps C.2.d, C.4 and C.5 of the procedure, there are no transitions t which can be made live, so deadlock prevention is impossible. ■

The following two results are very important in our development. Their proofs are more involved and so are included in the appendix.

Lemma 5.1 *Assume that no failure to transform a constraint to an admissible form occurs at any step C.2.b and for all minimal active siphons S controlled by the procedure the enforced constraint has the form (2) with α_p positive integers. Let S be an active siphon of \mathcal{N}_{i+1} , $i \geq 1$, which does not appear in \mathcal{N}_i . Let μ_{i+1} be a valid marking of \mathcal{N}_{i+1} such that S is empty and μ_i be μ_{i+1} restricted to \mathcal{N}_i . Let t_s be a transition of \mathcal{N}_i with the property that there is a transition $t \in S \bullet$ of \mathcal{N}_{i+1} such that $t_s = t$ or t_s is split in \mathcal{N}_{i+1} and t appears in a transition replacing sequence $\sigma_{i,i+1}(t_s)$. If $\exists \mu, \mu_s \in \mathcal{R}(\mathcal{N}_i, \mu_i)$ such that $\mu|t_s > \mu_s$, then (\mathcal{N}_i, μ_s) has at least one empty active siphon.*

Theorem 5.2 *In the assumptions of Lemma 5.1, the supervisor produced by the deadlock prevention method is at least as permissive as any of the supervisors subject to the same initial constraints (if any initial constraints are given) and which enforce that all transitions of the target Petri net which appear in the maximal active subnet are live, if such supervisors exists.*

Theorem 5.2 states that the supervisor provided by the procedure is at least as permissive as any supervisor which enforces all transitions of the maximal active subnet to be live in the target net. The comparison assumes that the other arbitrary supervisors are subject to the same initial constraints. Note also that Theorem 5.2 always applies to Petri nets with controllable and observable transitions. In order to apply to Petri nets with uncontrollable and unobservable transitions, the theorem requires that the admissible constraints (2) have positive coefficients rather than nonnegative. In the following corollary note that if liveness enforcing supervisors exist, then the target Petri net is repetitive.

Corollary 5.1 *In Theorem 5.2, the deadlock prevention procedure provides a supervisor at least as permissive as any liveness enforcing supervisor (subject to the same initial constraints), if any such supervisor exists.*

The procedure can be modified to guarantee termination. In what follows we consider the following modification: constraints of the form $\sum_{p \in S \cap R} \alpha_p \mu(p) \geq 1$ are enforced instead of $\sum_{p \in S} \alpha_p \mu(p) \geq 1$, where R is the set of the places

which have not been obtained by transition split (that is, the places of the target Petri net and the control places.) This modification may cause Theorem 5.2 not to be always applicable, but Theorem 5.1 still applies. For this modification we have the following termination result:

Theorem 5.3 *Let \mathcal{N}_0 be a Petri net and (L_I, b_I) be a set of constraints $L_I \mu \geq b_I$, $\mu \geq 0$, with bounded feasible region. Then the modified deadlock prevention procedure terminates if started with initial constraints (L_I, b_I) .*

Proof: Let $L'_0 \mu \geq b'_0$ be the form of $L_0 \mu \geq b_0$ after \mathcal{N}_0 is transformed to be PT-ordinary (step A of the procedure); this form applies for all \mathcal{N}_j , $j \geq 1$. By construction, since the feasible set of $L_I \mu \geq b_I$ is bounded (and so finite), so is the feasible set of $L'_0 \mu \geq b'_0$ (let it be \mathcal{M}_R). The modification of the procedure insures that all constraints associated to adding control places are only expressed in terms of the markings of the places of the target net \mathcal{N}_0 ; the marking of the places of the split replacements is never taken in account. So each time a new constraint is added to (L, b) or (L_0, b_0) , at least one new marking of \mathcal{M}_R is forbidden, because the constraint is not new unless the siphon generating it is uncontrolled (refer to the steps C.1 and C.2 of the procedure). Because \mathcal{M}_R is finite, after a finite number of iterations all new active siphons (if any) are controlled, and so the procedure terminates. ■

The usage of the modified procedure can be summarized as follows:

- Find a set of constrains $L_I \mu \geq b_I$ with bounded feasible set \mathcal{F} such that for all initial markings μ_0 of interest for \mathcal{N} : $\mathcal{R}(\mathcal{N}, \mu_0) \subseteq \mathcal{F}$. Let \mathcal{M}_I be the set of initial markings of interest.
- Use the procedure with the modification above and initial constraints (L_0, b_0) which equal (L_I, b_I) .
- The supervisor can be used for the initial markings $\mu_0 \in \mathcal{M}_I$ which satisfy $L \mu_0 \geq b$ and $L_0 \mu_0 \geq b_0$, where (L, b) and (L_0, b_0) are the two sets of constraints generated by the procedure.

VI. CONCLUSION

This paper has introduced a new deadlock prevention procedure. The performance of the procedure is formally proved. The procedure is effective for Petri net structures which may be generalized, with uncontrollable and unobservable transitions, nonrepetitive and unbounded. The initial marking is not required, instead the initial markings for which deadlock is prevented are characterized by a set of linear inequalities. Our approach to deadlock prevention has been implemented in software that performs automated synthesis of deadlock prevention supervisors, and is available from the authors.

APPENDIX

I. ADDITIONAL PROOFS

A. Proof of Lemma 5.1

Proof: Let \mathcal{C} be the set of control places added in the iteration i and P_R the set of places resulted through transition split in the iteration i : $P_R = P_{i+1} \setminus (P_i \cup \mathcal{C})$. Let σ be the firing sequence that was used to reach μ : $\mu_i[\sigma > \mu$. Consider firing σ in (\mathcal{N}_i, μ_i) and $\sigma' = \sigma_{i,i+1}(\sigma)$ in $(\mathcal{N}_{i+1}, \mu_{i+1})$. The only reason for σ' not to be enabled in \mathcal{N}_{i+1} by μ_{i+1} is that a control place prevents it.

If σ' is not enabled, $\sigma = \sigma_1 t_1 \sigma_2$, $\mu_i[\sigma_1 > \mu_i$, $\mu_{i+1}[\sigma_{i,i+1}(\sigma_1) > \mu'_{i+1}$, μ_1 enables t_1 , but μ'_1 does not enable $\sigma_{i,i+1}(t_1)$. This corresponds to the following: \mathcal{N}_i has an active siphon S_1 that is controlled in \mathcal{N}_{i+1} with C_1 ; when C_1 was added, $t_1 \in C_1 \bullet$, and if $W(C_1, t_1) > 1$, t_1 was split in step III-D of iteration i in $\sigma_{i,i+1}(t_1)$, or if $W(C_1, t_1) = 1$, $\sigma_{i,i+1}(t_1) = t_1$. So $t_1 \in S_1 \bullet$, and since t_1 is not allowed by C_1 to fire from μ_1 , it means that firing it would make S_1 empty. Since t_1 is fired in the sequence $\sigma = \sigma_1 t_1 \sigma_2$, after σ is fired, S_1 is an empty active siphon in (\mathcal{N}_i, μ_s) .

If σ' is enabled by μ_{i+1} , let μ' be the marking reached: $\mu_{i+1}[\sigma' > \mu'$. Because σ' may contain only entire replacement sequences of split transitions and μ_{i+1} is a valid marking (which implies $\mu_{i+1}(p) = 0 \forall p \in P_R$), $\mu'(p) = 0 \forall p \in P_R$. Also, μ_{i+1} and μ_i are equivalent and $\sigma' = \sigma_{i,i+1}(\sigma)$, therefore $\mu(p) = \mu'(p) \forall p \in P_i$. Because S is a siphon, S empty for μ_{i+1} implies S empty for all reachable markings, and so for μ' too. There are two cases: (a) t_s is not split in \mathcal{N}_{i+1} and (b) t_s is split.

(a) If t_s is not split, $\bullet t_s \cap P_R = \emptyset$. Further on, μ enables t_s in \mathcal{N}_i but μ' does not enable t_s in \mathcal{N}_{i+1} , so in \mathcal{N}_{i+1} , $\bullet t_s \cap \mathcal{C} \neq \emptyset$ and there is $C \in \bullet t_s \cap \mathcal{C}$ such that $\mu'(C) = 0$. Let S_C be the active siphon of \mathcal{N}_i controlled by C . t_s was not split, so $W(C, t_s)$ was 1; t_s enabled by μ , $\mu'(C) = 0$ and $t_s \in C \bullet \Rightarrow t_s \in (S_C \bullet) \setminus (\bullet S_C)$. Since $S_C \subseteq P_i$ and $\mu'(C) = 0$, $\sum_{p \in S_C} \mu(p) = 1$. Because t_s is enabled by μ , firing t_s empties S_C , so (\mathcal{N}_i, μ_s) has an empty active siphon.

(b) If t_s was split, then t_s was connected to one or more of the control places C of \mathcal{C} , for only transitions connected to such places are split. (This is so because for all $i \geq 1$ \mathcal{N}_i is PT-ordinary, and hence only the new added control places can make the Petri net not to be PT-ordinary.) Let \mathcal{C}_S be the set of control places added to $\bullet t_s$ in the iteration i . By recalling the split transition operation, it is easy to notice that $t \in S \bullet$ implies $\exists C \in \mathcal{C}_S$ such that $C \in S$. Let S_C be the active siphon controlled by C . Since $C \in S$ and S is empty, $\sum_{p \in S_C} \mu(p) = 1$. Since before the split of t_s $C \in \bullet t_s$, firing t_s in \mathcal{N}_i reduces the marking of S_C , and the total marking of S_C is one, S_C becomes empty. ■

B. Proof of Theorem 5.2

Proof: Let \mathcal{S} be the set of supervisors satisfying the initial constraints, which also enforce that all transitions which appear in the maximal active subnet are live in the target Petri net. Note that when we compare our procedure

to other supervisor we assume an initial marking for which that supervisor is defined: we do not require the supervisors in \mathcal{S} to be defined for all initial markings for which the supervisor given by our procedure is defined.

We first consider the case when there are no initial constraints. The proof is by contradiction. It shows that any marking forbidden by the deadlock prevention method also is forbidden by any supervisor in \mathcal{S} . Recall that our procedure forbids markings which will produce an empty active siphon in an \mathcal{N}_k for some k .

Let $\mu^{(1)}$ be a marking of \mathcal{N}_0 and $\mu_k^{(1)}$ the equivalent marking in \mathcal{N}_k . Suppose that for the marking $\mu_k^{(1)}$ there is an empty active siphon S_k in \mathcal{N}_k . Because $\mu_k^{(1)}$ is valid, S_k is a new siphon which does not appear in \mathcal{N}_{k-1} ; $\mu^{(1)}$ is forbidden by iteration k , which adds the constraint that S_k be controlled. Assume that $\mu^{(1)}$ is not forbidden by some supervisor enforcing in \mathcal{N}_0 that all transitions of the active subnet are live, and so there is an infinite firing sequence σ enabled by $\mu^{(1)}$ such that every transition of \mathcal{N}_0^A appears infinitely often in σ . According to Lemma 5.1, there is a transition t'_{k-1} of \mathcal{N}_{k-1} such that in any possible firing sequence, after t'_{k-1} fires in \mathcal{N}_{k-1} , there is an empty active siphon S_{k-1} of \mathcal{N}_{k-1} . Let $t_{k-1} \in T_0$ such that t'_{k-1} appears in $\sigma_{0,k-1}(t_{k-1})$. Let $\mu^{(2)}$ be the marking of \mathcal{N}_0 that appears while σ is fired, immediately after t_{k-1} fires for the first time. Also, let σ_1 be the subsequence of σ that was fired so far, that is $\mu^{(1)}[\sigma_1 > \mu^{(2)}$. Let $i \geq 0$ be the largest integer such that $\mu_i^{(2)}$ is a valid marking of \mathcal{N}_i and the restriction of $\mu_i^{(2)}$ to \mathcal{N}_0 is $\mu^{(2)}$. By Lemma 5.1, $i \leq k-1$. Indeed, if σ_1 is allowed to fire in \mathcal{N}_{k-1} , there is an empty siphon S_{k-1} for the marking $\mu_{k-1}^{(2)}$, but there is no valid marking of \mathcal{N}_k such that S_{k-1} is empty. Now, the fact that $\mu^{(2)}$ has an equivalent marking $\mu_i^{(2)}$ in \mathcal{N}_i but not in \mathcal{N}_{i+1} shows that there is an empty active siphon S_i in \mathcal{N}_i and that S_i does not appear in \mathcal{N}_{i-1} . Further on, the same idea as before is used, that a transition t_{i-1} with the same property as t_{k-1} exists, and following this idea, an index $j \leq i-1$ is found such that for the marking $\mu^{(3)}$ of \mathcal{N}_0 there is an empty active siphon in \mathcal{N}_{j-1} . This procedure is repeated and finally two cases may appear (Lemma 5.1 applies for $i > 0$ only) after the first n transitions of σ are fired, where n is a finite number. Let σ_p denote the sequence that enumerates the first n transitions of σ , and let $\mu^{(p)}$ be the marking reached by firing σ_p (that is, $\mu^{(1)}[\sigma_p > \mu^{(p)}$) and $\mu_1^{(p)}$ the valid marking of \mathcal{N}_1 which restricted to \mathcal{N}_0 is $\mu^{(p)}$. Then (a) there is an empty active siphon in $(\mathcal{N}_0, \mu^{(p)})$ or (b) there is an empty active siphon in $(\mathcal{N}_1, \mu_1^{(p)})$. Case (a) contradicts the fact that every transition appears infinitely often in σ and $\mu^{(1)}$ enables σ , since after n firings none of the transitions in the postset of the empty siphon may fire again. Case (b) leads to the same type of contradiction, because the sequence $\sigma' = \sigma_{0,1}(\sigma)$ is enabled by $\mu_1^{(1)}$, where $\mu_1^{(1)}$ is the equivalent marking of $\mu^{(1)}$ in \mathcal{N}_1 , and by construction every transition of \mathcal{N}_1^A appears infinitely often in σ' .

The case when there are initial constraints is similar to

the case when there are no such constraints if the procedure is never in the situation that a constraint at step C.2.d of the procedure is infeasible. If infeasibilities at some steps C.2.d occur, consider the first occurrence: there is an active siphon S which must be empty for all valid markings, in order not to have a conflict with the initial constraints. (In such a situation, being unable to control S , the procedure shrinks the active subnet such that S is no longer an active siphon.) Then, by the first part of the proof, there are no supervisors in \mathcal{S} . (\mathcal{S} is empty, as the initial constraints conflict with the requirement that the transitions of the maximal active subnet are live.) ■

REFERENCES

- [1] Barkaoui, K., I. Abdallah, "Deadlock Avoidance in FMS Based on Structural Theory of Petri Nets," *IEEE Symp. on Emerging Technologies and Factory Automation* 1995.
- [2] Barkaoui, K., J.-F. Pradat-Peyre, "On Liveness and Controlled Siphons in Petri Nets," in *Application and Theory of Petri Nets*, Springer Verlag, pp. 57-72, 1996.
- [3] Ezpeleta J., J. Colom, J. Martinez, "A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems," *IEEE Trans. on Rob. and Autom.*, vol 11, pp. 173-184, 1995.
- [4] Giua A., F. DiCesare, M. Silva, "Generalized Mutual exclusion Constraints on Nets with Uncontrollable Transitions," in *Proc. of the IEEE Intern. Conf. on Systems, Man and Cybernetics*, pp. 974-979, 1992.
- [5] Iordache M., J. Moody, P. Antsaklis "A Method for the Synthesis of Deadlock Prevention Controllers in Systems Modeled by Petri Nets," in *Proc. 2000 American Control Conference*.
- [6] Iordache M., *Automated Synthesis of Deadlock Prevention Supervisors Using Petri Nets*, Technical Report of the ISIS Group, ISIS-2000-003, University of Notre Dame, 2000.
- [7] Lautenbach K., H. Ridder, "The Linear Algebra of Deadlock Avoidance — A Petri Net Approach," Research Report at Institute for Computer Science, University of Koblenz, 1996.
- [8] Moody, J., P. Antsaklis, *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers, 1998.
- [9] Moody, J., P. Antsaklis, "Deadlock Avoidance Using the Supervisory Enforcement of Linear State Constraints on Petri Net Plants", Technical report, Univ. of Notre Dame, 1998.
- [10] Moody, J., P. Antsaklis, "Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions," in *IEEE Trans. Automat. Contr.*, vol. 45, pp. 462-476, 2000.
- [11] Murata, T. "Petri Nets: Properties, Analysis and Applications," in *Proc. of the IEEE*, vol. 77, pp. 541-580, 1989.
- [12] Reisig, W. *Petri Nets* Springer Verlag, 1985.
- [13] Sreenivas R., "On the Existence of Supervisory Policies that Enforce Liveness in Discrete Event Systems Modeled by Controlled Petri Nets," in *IEEE Trans. Automat. Contr.*, vol. 42, pp. 928-945, 1997.
- [14] Yamalidou K., J. Moody, M. Lemmon, P. Antsaklis, "Feedback control of Petri nets based on place invariants," in *Automatica*, vol. 32, pp 15-28, 1996.