Proceedings of the American Control Conference
Chicago, Illinois • June 2000

A METHOD FOR THE SYNTHESIS OF DEADLOCK PREVENTION CONTROLLERS IN SYSTEMS MODELED BY PETRI NETS

Marian V. Iordache*, John O. Moody**, Panos J. Antsaklis*
*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556
**Lockheed Martin Federal Systems, 1801 State Rt.17C, MD 0210, Owego NY 13827-3998

Abstract

Given an arbitrary Petri net structure, the deadlock prevention procedure presented here determines a set of linear inequalities on the marking of a Petri net. 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. The procedure is illustrated using an example from flexible manufacturing.

1 Introduction

Deadlock is an undesirable phenomenon that may occur in systems that contain components running in parallel and sharing common resources. A system is deadlocked when, due to mutual interdependencies and reliance on shared resources that can not be freed, no further actions can be taken by the system. *Deadlock prevention* differs from *liveness-insurance*: when the liveness of a system is guaranteed, all actions that a system can perform may be repeated infinitely often. Deadlock prevention insures that at least some subset of the system's actions may be repeated, but not necessarily all. Deadlock prevention may be applied to any system in which liveness can be guaranteed, however it is not possible to insure liveness for every system that can be made deadlock-free. The procedure presented here can be computationally expensive, however, all computations are performed off-line. This differentiates the technique from deadlock *avoidance* strategies that perform potentially expensive computations while the system is in operation. A controller resulting from our deadlock *prevention* method requires very little in terms of computational resources at run time.

Issues regarding conflict, synchronization, and concurrency naturally arise during the study of deadlock. These properties make the Petri net a particularly useful formalism for modeling systems susceptible to deadlock. 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 approach may be generalized to handle timed Petri nets or plants that include uncontrollable or unobservable transitions using techniques such as those in [6]. The method presents the conditions necessary to insure deadlock freedom as a set of linear integer inequalities. This formalism 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.

Deadlock prevention methods rely on structural properties of the net. Deadlock in Petri nets is related to *siphons* (see section 2.1 and [2]). Among deadlock prevention papers, [3] and [5] use *control places* to supervise the net, as in our approach. Control places are also used in [1]. The deadlock prevention method of [3] requires that the target Petri nets are *ordinary* and *conservative* and enforces *liveness* rather than deadlock prevention. The advantages of the method of [3] are simplicity and guaranteed success. The disadvantages are the assumptions made on the Petri net structure and that the method can be restrictive. The deadlock prevention method of [5] is intended for bounded Petri nets. It effectively handles nonordinary Petri nets. However, the approach in [5] is not effective for *nonrepetitive* Petri nets. Another problem is that [5] cannot *guarantee* deadlock prevention since it does not detect the case when the siphon supervision enforced by a control place is disabled by the transformation to ordinary Petri nets.

The new procedure we introduce is appropriate for use on nonrepetitive systems, in which liveness cannot be enforced under any circumstances. When the procedure is applied to repetitive systems, complete system liveness may well be the result. 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 is a *maximally permissive* liveness supervisor.

The main contributions of this paper are a methodology that adds significant improvements to previous methodologies and the formulation and proof of performance results. In Theorem 4.1 we give sufficient conditions that the procedure prevents deadlock for all initial markings that satisfy a linear matrix inequality. In Theorem 4.2 we show that the procedure is no more restrictive than any liveness enforcing supervisors. For a complete treatment, the reader is referred to the full-length paper [4], in which we provide the proofs, other new theoretical results, and detailed analysis of the procedure.

Section 2 presents several preliminary Petri net results; section 3 introduces the procedure. Properties of the method are given in section 4 and a flexible manufacturing example in section 5.

2 Preliminaries

2.1 Deadlock and Siphons

In this paper we assume that the reader is familiar with the fundamentals of Petri nets; see for instance [7] and [8]. Let $\mathcal{N} = (P, T, F, W)$ be a Petri net structure, where P is the set of places, T the set of transitions, F the set of transition arcs and $W : F \rightarrow \mathbb{N} \setminus \{0\}$ is the weight function. $S \subseteq P$ is a **siphon** of $\mathcal{N}$ if $S \neq \emptyset$ and $\bullet S \subseteq S \bullet$. S is a **minimal siphon** if there is no other siphon S' such that $S' \subset S$. The siphon S is empty for the current marking μ if $\mu(p) = 0\ \forall p \in S$. The Petri net $\mathcal{N}$ is **ordinary** if $\forall f \in F : W(f) = 1$. We say that $\mathcal{N}$ is **PT-ordinary** if $\forall f \in F \cap (P \times T)$: $W(f) = 1$. A well known necessary condition for an ordinary Petri net to be in deadlock is that it contains an empty siphon [8]. This result can be easily extended to PT-ordinary Petri nets.

Proposition 2.1 *A deadlocked PT-ordinary Petri net contains at least one empty siphon.*

A siphon S is **controlled** with respect to a set of initial markings $\mathcal{M}_I$ of a Petri net if for all reachable markings S is never empty. A siphon containing a controlled siphon is controlled. Hence a PT-ordinary Petri net such that all minimal siphons are controlled is deadlock-free by Proposition 2.1.

2.2 Supervisors Based on Place Invariants

Yamalidou and Moody have proposed in [6] and [9] a method to synthesize supervisors enforcing linear marking inequalities. The form of the inequalities is $L\mu \leq b$, where μ is the marking. The construction is summarized in the following theorem:

Theorem 2.1 ([6] and [9]) *Let a plant Petri net with controllable and observable transitions, incidence matrix D_p and initial marking μ_{p0} be given. A set of n_c linear constraints $L\mu_p \leq b$ are to be imposed. If $b - L\mu_{p0} \geq 0$ then a Petri net controller (supervisor) with incidence matrix $D_c = -LD_p$ and initial marking $\mu_{c0} = b - L\mu_{p0}$ enforces the constraint $L\mu_p \leq b$ when included in the closed loop system $D = [D_p^T, D_c^T]^T$. Furthermore, the supervision is maximally permissive.*

Because $D_c = -LD_p$, every row of $[L, I]$ is a place invariant of the incidence matrix D of the closed loop. A quality of this technique is that the closed loop (i.e. supervised) system remains a Petri net.

The condition that a siphon S is controlled can be written: $\sum\limits_{p \in S} \mu(p) \geq 1$. Hence we can use supervision based on place invariants to control siphons. The method applied to a siphon S yields an additional place C, which we call **control place**. The supervision of S creates the place invariant described by the equation $\mu(C) = \sum\limits_{p \in S} \mu(p) - 1$.

2.3 A Transformation of Petri Nets to PT-ordinary Petri Nets

Because Proposition 2.1 applies to PT-ordinary Petri nets, we are interested in using a transformation of Petri nets to PT-ordinary Petri nets. The following procedure is a slightly modified form of the transformation of Lautenbach and Ridder [5].

Each transition t_j such that $W(p, t_j) > 1$, where $p \in \bullet t_j$, is **split** (decomposed) in m transitions: $t_{j,1}$, $t_{j,2}$, $\ldots$ $t_{j,m}$, where $m = \max\{W(p, t_j) : (p, t_j) \in F\}$. Also, $m - 1$ new places are added: $p_{j,1}, p_{j,2}, \ldots p_{j,m-1}$. The transition split operation replaces t_j with the transitions $t_{j,i}$ and the places

$p_{j,i}$, interconnected as follows:

- $\bullet p_{j,i} = t_{j,i}$, $t_{j,i}\bullet = p_{j,i}$ and $p_{j,i}\bullet = t_{j,i+1}$, for $i = 1 \ldots m-1$
- $\bullet t_{j,i} = \{p \in \bullet t_j : W(p, t_j) \geq i\}$, for $i = 2 \ldots m$
- $\bullet t_{j,1} = \bullet t_j$ and $t_{j,m}\bullet = t_j\bullet$

The weights of the new transition arcs are all 1, except for the arcs of the form $(t_{j,m}, p)$, for which the weight is $W'(t_{j,m}, p) = W(t_j, p)$.

3 The Deadlock Prevention Method

3.1 Introduction to the Method

Given a Petri net $\mathcal{N}_0$, the method generates a sequence of PT-ordinary Petri nets, $\mathcal{N}_1$, $\mathcal{N}_2$, $\ldots$ $\mathcal{N}_k$, increasingly enhanced for deadlock prevention. $\mathcal{N}_1$ is $\mathcal{N}_0$ transformed to be PT-ordinary. The other Petri nets are largely obtained as follows: in each iteration i the new minimal siphons of $\mathcal{N}_i$ are controlled, and then, if needed, transitions are split; the resulting PT-ordinary net is $\mathcal{N}_{i+1}$. Recall, for each controlled 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 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 in constraints in terms of the markings of $\mathcal{N}_0$; these constraints define the supervisor for deadlock prevention in $\mathcal{N}_0$.

The procedure can work as outlined above when $\mathcal{N}_0$ is a repetitive Petri net. In order to be able to approach Petri nets which are not repetitive, that is Petri nets whith structurally nonlive transitions, the enhancements of section 3.2 have been made. In section 3.3 we show how constraints are translated for the target Petri net $\mathcal{N}_0$.

3.2 Extensions to Deal with Source Places

Note that the source places of a net are minimal siphons. They cannot be controlled with the approach of section 2.2, because the control place which results is another source place, which is a new minimal siphon. So, the procedure would not converge. If source places are ignored, examples can be found to show that the procedure would not prevent deadlock. Note that source places can appear during iterations even if $\mathcal{N}_0$, the target net, has no source places. To solve this problem, the procedure considers for control only the siphons which belong to a subnet of the Petri net. The **active subnet** $\mathcal{N}^A$ of the Petri net $\mathcal{N}$ is obtained by repeatedly removing all source places and all transitions connected to the source places, until no source places remain in the net. The procedure is modified to only consider the siphons of $\mathcal{N}_1^A$, $\mathcal{N}_2^A$, $\ldots$ $\mathcal{N}_k^A$.

3.3 Implicit Inequalities

Occasionally a new constraint introduced by the procedure may already be implicitly enforced through previous constraints or through the intrinsic structural properties of the net. Determining that a new inequality is implicitly enforced can be done through analysis of the plant's original place-invariants and the place-invariants imposed by previous constraints. Implicit inequalities are removed from the list of deadlock-avoidance inequalities to reduce the complexity of the resulting controller.

Assume that we are in iteration number i. Let P_R be the set of places resulted through transition splits. Consider the markings such that $\mu(p) = 0$ $\forall p \in P_R$. The place invariants which appear when a siphon S is controlled with the control place C are described by $\mu(C) = \sum_{p\in S} \mu(p) - 1$. The equation enforces the requirement that $\sum_{p\in S} \mu(p) \geq 1$. Because S may contain control places added in previous iterations, which are described by similar equations, after repeated substitutions, we get the requirement on S expressed in the form $l^T\mu \geq c$, where the vector l may have nonzero entries only for places of the target net $\mathcal{N}_0$. All constraints $l^T\mu \geq c$ can be written compactly as a matrix inequality $L\mu \geq b$.

It may be possible that a siphon S, due to structural properties of the net, cannot become empty if initially S is not empty. This case corresponds to $C\bullet \subseteq \bullet S$. Then C is unnecessary, and we only need to require that $l^T\mu_0 \geq c$ for all initial markings μ_0. All constraints for such siphons are written compactly as $L_0\mu_0 \geq b_0$.

3.4 The Deadlock Prevention Procedure

The purpose of the procedure is to specify the supervisor of the target Petri net $\mathcal{N}_0$ in terms of the marking constraints $L\mu \geq b$ and $L_0\mu \geq b_0$. The procedure starts with an empty set of constraints (L, b). The form of each iteration $i \geq 1$ is

1. If no new minimal siphon of $\mathcal{N}_i^A$ is found, the procedure terminates.

2. For every new minimal siphon S of $\mathcal{N}_i^A$, the control method of section 2.2 is used to enforce $\sum_{p\in S} \mu(p) \geq 1$ in $\mathcal{N}_i$. Let C be the control place which would result and $l^T\mu \geq c$ the corresponding inequality written in terms of the marking of the

places of $\mathcal{N}_0$. There are two cases:

(a) $C\bullet \subseteq \bullet S$. Then C is not added to $\mathcal{N}_i$ and the linear constraint (l, c) is included in (L_0, b_0).

(b) $C\bullet \not\subseteq \bullet S$. Then the place C is added to $\mathcal{N}_i$ according to the approach of section 2.2 and the linear constraint (l, c) is included in (L, b). $\mathcal{N}_i^A$ is updated as follows: C is added to $\mathcal{N}_i^A$; for each transition t of $\mathcal{N}_i^A$ connected to C in $\mathcal{N}_i$, the arc connecting t to C in $\mathcal{N}_i$ is copied in $\mathcal{N}_i^A$, together with its weight.

3. The active subnet $\mathcal{N}_i^A$ is updated as in section 3.2 (source places may appear in step 2.) The control of a siphon S performed in step 2(b) is said to **fail** if in the updated $\mathcal{N}_i^A$ $\exists t \in C\bullet \setminus \bullet S$ such that the arc (C, t) has a weight greater than one.

4. If the active subnet no longer is PT-ordinary, the transitions of the active subnet which do not comply with this requirement are split in both $\mathcal{N}_i$ and $\mathcal{N}_i^A$. The final nets of iteration i are denoted by $\mathcal{N}_{i+1}^A$ and $\mathcal{N}_{i+1}$.

After the iterative process terminates, a method that removes redundant constraints may be used to simplify (L, b). The inequalities given by (L, b) (in terms of markings of the original net $\mathcal{N}_0$) are enforced on $\mathcal{N}_0$ with the invariant based methodology of section 2.2. For all initial markings μ_0, such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, deadlock prevention in the closed loop Petri net is guaranteed in the conditions of Theorem 4.1.

3.5 Illustrative Example

Consider the Petri net of Figure 1(a). No source places are present, so the active subnet is equal to the PT-transformed net (Figure 1(d)). In the first iteration, there is a single minimal siphon of $\mathcal{N}_1^A$: $\{p_1, p_2, p_3\}$. A control place C_1 is added to the total net (Figure 1(e)); C_1 is a source place. The resulting active subnet is shown in Figure 1(c). The inequality associated with C_1 is $\mu(p_1)+\mu(p_2)+\mu(p_3) \geq 1$, so at the end of the iteration $L = [1, 1, 1]$ and $b = 1$. In the second iteration the active subnet has a single siphon, $\{p_1, p_2\}$, and a control place C_2 is added. The matrices L and b become

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

The procedure terminates at the third iteration, since there is no new minimal siphon in the active subnet. There is one redundant constraint in (L, b). That constraint is removed, and the supervisor is built according to $L = [1, 1, 0]$ and $b = 1$. The supervised net is shown in Figure 1(b).

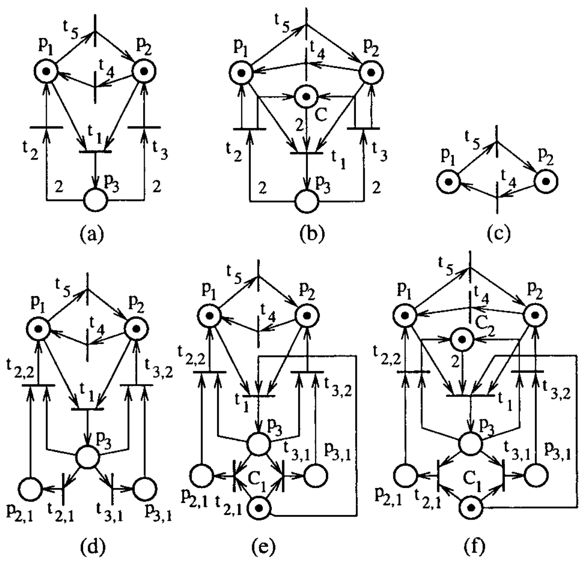

Figure 1: The example of section 3.5: (a) $\mathcal{N}_0$; (b) the final Petri net supervised for deadlock-freedom; (c) $\mathcal{N}_2^A$, the same as $\mathcal{N}_3^A$; (d) $\mathcal{N}_1^A$, the same as $\mathcal{N}_1$; (e) $\mathcal{N}_2$; (f) $\mathcal{N}_3$.

4 Properties

The first theorem gives a sufficient condition for the procedure to prevent deadlock.

Theorem 4.1 [4] *Assume that the procedure terminates after $k-1$ iterations, $\mathcal{N}_k^A$ is nonempty and no siphon control failure occurred in the step 3 of any iteration. 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 $L_0\mu_0 \geq b_0$.*

The next theorem states that the set of markings forbidden by the supervisor generated by the procedure is a subset of the set of markings forbidden by any liveness enforcing supervisor, if any exist. It also shows that in the cases when the supervisor enforces liveness [4], it is maximally permissive.

Theorem 4.2 [4] *The procedure provides a supervisor at least as permissive as any liveness enforcing supervisor, if any exists.*

5 Application: A Flexible Manufacturing System

Consider the flexible manufacturing system model shown in figure 2. The places p_4, p_5, p_6, p_{15} and p_{16} correspond to 5 different resources (machine types);

their marking corresponds to the number of available resources. Resources from p_4 can be used in the manufacturing cells p_1 and p_7, from p_5 in p_2 and p_8, from p_6 in p_3 and p_9, from p_{15} in p_{11} and p_{17} and from p_{16} in p_{12} and p_{18}. A working process in p_{11} requires 2 units of the resource corresponding to p_{15}. The parts that are to be processed enter the points A and B; a part in process corresponds to a token. Parts entering A may leave the production cell in the points N or O; parts entering B may leave the production cell in M or P.

The production cell is designed to be used in sessions of finite duration. The resources of p_{15} may require maintenance during a manufacturing session for performance improvement. Such a resource can go through the maintenance process, and this is modeled by firing t_{14} and then t_{15}. However, this operation takes much more time than to process a part. A resource of type p_{15} which requires maintenance would be good enough to be used as a resource of type p_4. Depending on the circumstances, it may be more efficient to use it as a resource of type p_4, rather than fixing it; this is modeled by firing t_9. Because very infrequently resources of type p_{15} require maintenance during a manufacturing session, a source place p_{14} is used, whose marking corresponds to an estimated upper bound of the number of maintenance requests that may occur during a manufacturing session.

The supervisor for deadlock prevention has been automatically generated in 3 iterations. Ten constraints were generated: 6 in (L, b) and 4 in (L_0, b_0). Therefore six control places were added. In this example the procedure performs very well: not only is deadlock prevented, but also all the transitions which can be made live are live. The example has been chosen such that none of the approaches in [1], [3] and [5] is applicable.

6 Conclusion

A deadlock prevention procedure for Petri nets is proposed, in which the supervisor is defined by a set of linear marking inequalities. Deadlock prevention is guaranteed in the conditions of Theorem 4.1. The permissivity of the supervisor is at least as good as that of any supervisor which enforces liveness. The procedure does not require the initial marking to be known; instead it characterizes the usable initial markings as the feasible region of a set of linear inequalities. A major advantage of this approach is that it is very general, being applicable to generalized Petri net structures.

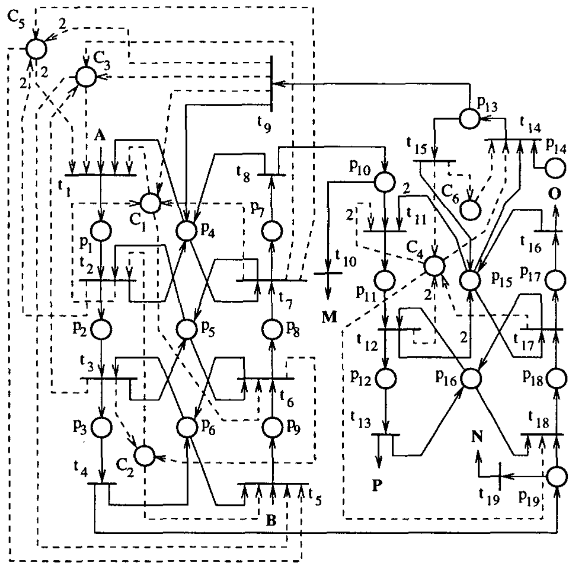

Figure 2: The structure of the Petri net model and of the supervisor

References

[1] Barkaoui, K., I. Abdallah, (1995) "Deadlock Avoidance in FMS Based on Structural Theory of Petri Nets," *IEEE Symposium on Emerging Technologies and Factory Automation.*

[2] Boer E., T. Murata, (1994) "Generating Basis Siphons and Traps of Petri Nets Usign the Sign Incidence Matrix," *IEEE Trans. on Circ. and Sys.*, 41(4).

[3] Ezpeleta J. et al (1995) "A Petri Net Based Deadlock Prevention Policy for FMS," *IEEE Trans. on Robotics and Automation*, 11(2).

[4] Iordache M., J. Moody, P. Antsaklis (1999) *A Method for Deadlock Prevention in Discrete Event Systems Using Petri Nets*, Technical Report of the ISIS Group, ISIS-99-006, University of Notre Dame, available at `http://www.nd.edu/~isis/tech.html`.

[5] Lautenbach K., H. Ridder, (1996) "The Linear Algebra of Deadlock Avoidance — A Petri Net Approach," Research Report at Institute for Computer Science, University of Koblenz, Germany.

[6] Moody, J., P. Antsaklis, (1998) *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers.

[7] Murata T. (1989) "Petri Nets: Properties, Analysis and Applications," in *Proc. of the IEEE*, 77(4).

[8] Reisig W. (1985) *Petri Nets*, Springer Verlag.

[9] Yamalidou K., J. Moody, M. Lemmon, P. Antsaklis, (1996) "Feedback control of Petri nets based on place invariants," in *Automatica*, 32(1) .