Marian V. Iordache, John O. Moody and Panos Antsaklis, "A Method for the Synthesis of Liveness Enforcing Supervisors in Petri Nets," Proceedings American Control Conference, pp. 4943-4948, Arlington, VA, June 24-27, 2001

Proceedings of the American Control Conference
Arlington, VA June 25-27, 2001

A Method for the Synthesis of Liveness Enforcing Supervisors in Petri Nets

Marian V. Iordache[1], John O. Moody[2] and Panos J. Antsaklis[1,3]

Abstract

Given an arbitrary Petri net structure which may have uncontrollable and unobservable transitions and may be unbounded, the procedure described in this paper generates a supervisor for liveness enforcement. The supervisor is specified as a conjunction of linear marking inequalities. For all initial markings satisfying the linear marking inequalities, the supervised Petri net is live. Moreover, the supervision is least restrictive in the fully controllable and observable case.

1 Introduction

Liveness in a Petri net means that for all markings reachable from an initial marking any transition can eventually be fired. This implies absence of deadlock. In general Petri nets may not be live. Our procedure finds a supervisor such that given a Petri net, the supervised Petri net is live for all initial markings satisfying a set of linear inequalities. For the Petri nets which do not need supervision for liveness, the procedure may still be used to generate the set of initial markings for which the Petri net is live.

The procedure is recursive. At every iteration it finds local deadlock situations, corrects them, then transforms the Petri net to a convenient form, to be an asymmetric choice ordinary Petri net. The supervisors are built according to the invariant based supervision defined in [8, 12]. This has the advantage that the supervisors are defined independently of the initial marking and the supervised Petri net can still be represented as a Petri net. It is a distinguished feature of our procedure that the supervisors it provides are not dependent on a single initial marking, but are valid for initial markings for which liveness can be enforced. When the Petri net structure is fully controllable/observable and our procedure terminates, the supervisor is maximally permissive. In this case for each initial marking for which the supervisor is defined there is no other liveness enforcing supervisor less restrictive, and for all markings for which the supervisor is not defined liveness enforcing is impossible. In principle a reachability graph could be used to generate a liveness enforcing supervisor when the initial marking is given and the Petri net is bounded. However we consider arbitrary Petri nets, which may be unbounded, and we characterize the set of markings for which liveness can be enforced. Thus the problem we solve cannot be solved with finite automaton techniques.

The procedure can be automatically performed by a computer. However it may not always terminate and each iteration may perform computationally expensive operations. Nevertheless, all computations are performed off-line. Thus the supervisors designed by the procedure are appropriate for real-time applications.

There are not many results in the literature about enforcing liveness in Petri nets. An unfolding based liveness enforcing approach for ordinary and n-safe Petri nets appears in [3]. A method for liveness enforcement in a class of conservative ordinary Petri nets has been given in [2]; the approach is not maximally permissive. A liveness enforcing approach for a restricted class of ordinary Petri nets is given in [10]. Another liveness enforcing approach appears in [11]; it is based on the coverability graph, and hence the initial marking is required. Our approach is most related to the deadlock prevention procedure we presented in [6], and its improvement in [4]. While our former procedure prevented deadlock but was not guaranteed to enforce liveness, the procedure of this paper is guaranteed to enforce liveness.

We begin by introducing in section 2 notations, definitions and the previous results which we use. Then in section 3 we define two transformations which we use in the liveness enforcement procedure, which is defined in section 4. We state our theoretical results in section 5.

2 Preliminaries

We assume that the reader is familiar with the usual Petri net notations and definitions; see for instance [9]. We denote a Petri net structure by $\mathcal{N} = (P, T, F, W)$, where P is the set of places, T the set of transitions, F the set of transition arcs and W the weight function. A

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

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

[3]The authors gratefully acknowledge the partial support of the National Science Foundation (ECS-9912458) and of the Army Research Office (DAAG55-98-1-0199).

firing sequence is a sequence of transitions. A firing sequence may be infinite. A Petri net $(\mathcal{N}, \mu_0)$ is said to be **live** if for all reachable markings μ and for all transitions t of $\mathcal{N}$ there is a firing sequence σ enabled by μ which includes t. We regard a **supervisory policy** as a map from a set of markings to the power set of the set of transitions. Thus, a supervisory policy associates to a marking a set of transitions which are allowed to be fired if enabled. We will say that a Petri net **can be made live** if there is a supervisory policy such that the supervised Petri net is live. A Petri net is **repetitive** [9] if a marking μ_0 and an infinite firing sequence σ enabled by μ_0 exist, such that every transition occurs infinitely often in σ. A Petri net can be made live only if it is repetitive. Note that it can easily be checked whether a Petri net is repetitive [4].

A Petri net is **ordinary** if $\forall f \in F : W(f) = 1$. We say that a Petri net is **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 with **asymmetric choice** if for all places p_1 and p_2 such that $p_1 \bullet \cap p_2 \bullet \neq \emptyset$: $p_1 \bullet \subseteq p_2 \bullet$ or $p_2 \bullet \subseteq p_1 \bullet$. The set $S \subseteq P$ 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** with respect to the current marking μ if $\mu(p) = 0\ \forall p \in S$. A siphon S is **controlled** [1] with respect to an initial marking if for all reachable markings S is never empty. The following theorem is a special case of a result in [1]:

Theorem 2.1 *A PT-ordinary asymmetric choice Petri net is live iff all minimal siphons are controlled.*

In this paper we use the supervisory technique of [8, 12]. Given a set of inequalities of the form $L\mu \geq b$, the construction of the supervisor enforcing it is outlined in the following theorem of [8, 12]

Theorem 2.2 *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 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.*

3 Petri Net Transformations

3.1 Transformation to PT-ordinary Petri Nets

We use a modified form of the transformation of [7], and we call it *PT-transformation*. Let $\mathcal{N} = (P, T, F, W)$ be a Petri net. The **PT-transformation** consists in *splitting* all transitions t such that $W(p,t) > 1$ for some $p \in \bullet t$. We define the **transition split** as follows. Given $t_j \in T$, let $n(t_j) = \max\{W(p,t_j) : (p,t_j) \in F, p \in P\}$. Then t_j is *split* in $m = n(t_j)$ transitions: $t_{j,0}, t_{j,1}, t_{j,2}, \ldots t_{j,m-1}$. Also, $m-1$ new places are generated: $p_{j,1}, p_{j,2}, \ldots p_{j,m-1}$, where:

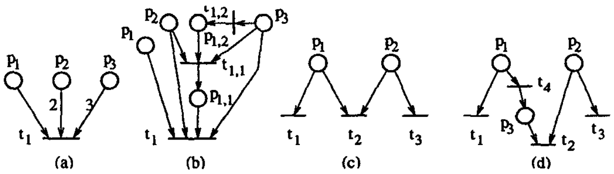

Figure 1: Illustration of the transformations: (a) initial configuration; (b) transition split for PT-transformation; (c) another initial configuration (d) transition split for AC- transformation.

a. $\bullet p_{j,i} = t_{j,i}$, $t_{j,i}\bullet = p_{j,i}$ and $p_{j,i}\bullet = t_{j,i-1}$, for $i = 1 \ldots m-1$

b. $\bullet t_{j,i} = \{p \in \bullet t_j : W(p,t_j) > i\}$, for $i = 0 \ldots m-1$

c. $t_{j,0}\bullet = t_j\bullet$

Note that t_j resembles $t_{j,0}$: $t_{j,0}$ has all the connections of t_j plus one additional transition arc. *After the split is performed, we denote $t_{j,0}$ by t_j.* In terms of marking inequalities, splitting t_j changes a marking inequality $\sum g_p\mu(p) \geq f$ to $\sum g'_p\mu(p) + \sum \beta_i\mu(p_{j,i}) \geq f$, where the latter is obtained by substituting for all $p \in \bullet t$

$$\mu(p) \longrightarrow \mu(p) + \sum_{i=1}^{v-1} i\mu(p_{j,v-i}) \tag{1}$$

and $v = W(p,t_j)$. We use the convention that a split transition t_j is also a transition of the PT-transformed net, since we denote $t_{j,0}$ by t_j. Firing an unsplit transition t_j in the original net corresponds to firing the same transition in the transformed net. Firing a split transition t_j in the original net corresponds to firing the sequence $t_{j,m} \ldots t_{j,1}, t_j$ in the transformed net. Figure 1(a-b) shows an example.

3.2 Transformation to Asymmetric Choice Petri Nets

Let $\mathcal{N} = (P, T, F, W)$ be a Petri net and $\mathcal{N}' = (P', T', F', W')$ be the transformed Petri net, where $P \subseteq P'$, $T \subseteq T'$. The idea of the transformation is as follows. Given the transition t, $p_i \in \bullet t$ and $p_j \in \bullet t$ such that $p_i\bullet \not\subseteq p_j\bullet$ and $p_j\bullet \not\subseteq p_i\bullet$, remove t from either the postset of p_i or that of p_j by adding a new place and a new transition. The idea is illustrated in Figure 1(c-d).

Algorithm of the AC-Transformation

Input: $\mathcal{N}$ and optionally $M \subseteq P$; the default value of M is $M = P$.

Output: $\mathcal{N}'$

Initialize $\mathcal{N}'$ to $\mathcal{N}$. (The algorithm takes the preset/postset with respect to $\mathcal{N}'$.)

For every $t \in T$ with $|\bullet t| > 1$ **do**

A. Construct $U = \{(p_i, p_j) \in P \times P : p_i \in \bullet t, p_j \in \bullet t, p_i\bullet \not\subseteq p_j \bullet \text{ and } p_j\bullet \not\subseteq p_i\bullet\}$.

B. **if** U is empty, **then** continue with the next iteration.

C. Let $Q := \emptyset$.

D. **For** every $(p_i, p_j) \in U$

D.1. A place $p \in \{p_i, p_j\} \cap M$ is selected. If two choices are possible:

D.1.a. $p = p_i$ (or $p = p_j$) if p_i (or p_j) has been previously selected for another element of U.

D.1.b. otherwise p is chosen such that p appears in other element of U. If both p_i and p_j satisfy this property, select $p \in \{p_i, p_j\}$ such that $|p\bullet| = \max\{|p_i\bullet|, |p_j\bullet|\}$.

D.1.c. if none of p_i and p_j appears in another element of U, select $p \in \{p_i, p_j\}$ such that $|p\bullet| = \max\{|p_i\bullet|, |p_j\bullet|\}$.

D.2. If a place p could be selected (i.e. if $\{p_i, p_j\} \cap M \neq \emptyset$) then $Q := Q \cup \{p\}$

E. **For** all $p \in Q$, delete from $\mathcal{N}'$ the transition arc (p, t) and add a new place p' and a new transition t' such that $\bullet t' = \{p\}$, $t'\bullet = \{p'\}$, $p'\bullet = \{t\}$, $W'(p, t') = W'(t', p') = 1$ and $W'(p', t) = W(p, t)$.

We call the transformation to asymmetric choice Petri nets **AC-transformation**. The operation in the step E is a **transition split**. The transition split of the AC-transformation is slightly different from the transition split of the PT-transformation. With regard to marking inequalities, when t is split into $\{t', t\}$, a marking inequality $\sum g_p \mu(p) \geq f$ is changed into $\sum g_p \mu(p) + g_{p_0} \mu(p_0') \geq f$, where $p_0 = \bullet t'$ and $p_0' = t'\bullet$.

4 The Liveness Enforcement Procedure

4.1 Introduction to the Method

Let $\mathcal{N}_0 = (P_0, T_0, F_0, W_0)$ be a Petri net structure. The liveness enforcement procedure generates marking constraints $L\mu \geq b$, $L \in \mathbb{N}^{n_c \times |P_0|}$ and $b \in \mathbb{N}^{n_c}$, such that when $\mathcal{N}_0$ is supervised such that $L\mu \geq b$ holds true, the supervised structure is live for all initial markings μ_0 satisfying $L\mu_0 \geq b$.

In order to generate $L\mu \geq b$, the procedure iteratively generates the asymmetric choice PT-ordinary Petri nets $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \ldots \mathcal{N}_k$. $\mathcal{N}_1$ is $\mathcal{N}_0$ transformed to be PT-ordinary and with asymmetric choice. For $i = 1 \ldots k$, let $L_i\mu \geq b_i$ be $L\mu \geq b$ at the iteration i, where μ is the marking of the places which are not control places; let μ_c be the marking of the control places. For $i = 1 \ldots k-1$, the uncontrolled new minimal siphons of $\mathcal{N}_i$ are controlled by adding new places to the net (section 4.2). (A minimal siphon S of $\mathcal{N}_i$ is **new** if $i = 1$ or if $i > 1$ and S is not a siphon of $\mathcal{N}_{i-1}$; S is **uncontrolled** if not controlled for μ such that $L_i\mu \geq b_i$, $\mu(p) = 0$ for $p \notin P_0$ and $\mu_c = L_i\mu - b_i$.) The obtained Petri net structure may no longer be PT-ordinary and with asymmetric choice and hence is transformed to be so; the result is $\mathcal{N}_{i+1}$. The procedure terminates for $i = k$ such that $\mathcal{N}_k$ has no uncontrolled new minimal siphons. Each time the procedure controls a siphon S, a new constraint of the form (2) is included in $L_i\mu \geq b_i$. The constraints $L_k\mu \geq b_k$ are translated to be written only in terms of the places of the target Petri net $\mathcal{N}_0$. The result is the final $L\mu \geq b$.

Section 4.2 describes the constraints $L\mu \geq b$. Section 4.3 considers transforming the constraints of the form (2) when dealing with Petri nets with uncontrollable and unobservable transitions. The procedure is stated in section 4.4 and examples are given in section 4.5.

4.2 Generating Constraints

The condition that a siphon S is controlled is

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

We use the approach of Theorem 2.2 to control S. This yields an additional place C called **control place**, and creates the place invariant described by the equation

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

For a control place C added in some iteration the final form of the invariant is not (3) but

$$\mu(C) + \sum_{p \in U} \beta_p \mu(p) = \sum_{p \in S} \mu(p) - 1 \tag{4}$$

due to the PT and AC-transformations performed at the end of the iteration. In (4) U is the set of additional places generated by the PT and AC transformations and $\beta_p \geq 0$. In general a siphon S may contain control places added in previous iterations; their markings satisfy equations (4). Then, by repeated substitutions, we get the requirement (2) on S expressed in the form $l^T\mu \geq c$, where the vector l has zero entries for control places. The constraint (2) written as $l^T\mu \geq c$ is included in $L\mu \geq b$. An important feature of our procedure is that the siphons of $\mathcal{N}_i$, $i \geq 1$, are siphons in $\mathcal{N}_{i+1}$, and they are controlled in $\mathcal{N}_{i+1}$. The PT and AC transformations performed in an iteration affect only the connections of the control places added in that iteration.Thus the constraints included in $L\mu \geq b$ in an iteration remain valid in the subsequent iterations.

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 as $L_0\mu_0 \geq b_0$ and they are not included in $L\mu \geq b$.

The constraints for $\mathcal{N}_0$ are obtained from that of $\mathcal{N}_k$ by removing the columns of L and L_0 which correspond to places not in $\mathcal{N}_0$.

4.3 Admissible Constraints

If the Petri net contains uncontrollable and unobservable transitions, it is necessary that the final constraints $L\mu \geq b$ of $\mathcal{N}_0$ are ***admissible*** [8]. That is, the supervisor enforcing $L\mu \geq b$ does not attempt to disable enabled uncontrollable transitions and to observe unobservable transitions. It has been noticed in [8] that if D_{uc} and D_{uo} represent the incidence matrix restricted to the set of uncontrollable and unobservable transitions, respectively, then the admissibility requirement is satisfied if $LD_{uc} \geq 0$ and $LD_{uo} = 0$.

In order to generate admissible constraints, when (2) would result in an inadmissible constraint, the procedure changes (2) to the form

$$\sum_{p \in S} \alpha_p \mu(p) \geq 1 \tag{5}$$

where $\alpha_p \geq 0$ are integers, at least two of α_p are nonzero and (5), expressed in the form $l^T\mu \geq c$ (section 4.2) and restricted to the places of $\mathcal{N}_0$, is admissible. We present the algorithm which transforms constraints (2) to admissible constraints of the form (5) in [4].

4.4 The Procedure for Liveness Enforcement

Input: The target Petri net $\mathcal{N}_0$.

Output: Two sets of constraints (L, b) and (L_0, b_0) (liveness is enforced in $(\mathcal{N}_0, \mu_0)$ supervised according to $L\mu \geq b$ for all initial markings μ_0 such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$)

A. $\mathcal{N}_0$ is transformed to be PT-ordinary and then to have asymmetric choice, as shown in the sections 3.1 and 3.2. The transformed net is $\mathcal{N}_1$. Let $i = 1$.

B. **For** $i \geq 1$ **do** (the initial Petri net of the iteration i is $\mathcal{N}_i$.)

B.1. If no new uncontrolled minimal siphon is found, the next step is C.

B.2. For every new uncontrolled minimal siphon S:

Let C be the control place which would result by controlling the siphon, and let $l^T\mu \geq c$ be (2) written in the form shown in section 4.2. First, the approach of section 4.2 is considered for the control of S through C.

B.2.a. If $C\bullet \subseteq \bullet S$, then S does not need supervision and C is not added to $\mathcal{N}_i$. The constraint (l, c) is added to (L_0, b_0).

B.2.b. If $C\bullet \not\subseteq \bullet S$, C is added by enforcing (2) in the Petri net (section 4.2), if (2) is admissible. If (2) is inadmissible, it is transformed to (5), and C is added to enforce (5). If the transformation to the form (5) is not possible, the procedure terminates, as it cannot enforce liveness.

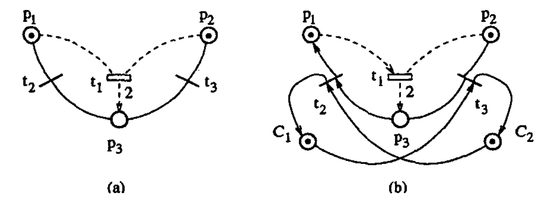

Figure 2: Illustrative example: (a) $\mathcal{N}_0$; (b) the Petri net supervised for liveness

B.3. If the Petri net is no longer PT-ordinary, the Petri net is PT-transformed (section 3.1.)

B.4. If the Petri net is no longer with asymmetric choice, the Petri net is AC-transformed (section 3.2), where the second argument M of the AC-transformation is taken to be the set of the control places added in the current iteration.

B.5. The matrices L and L_0 are enhanced with new columns, each column corresponding to one new place resulted in the steps B.3 and B.4.

B.6. The final net of the iteration i is denoted $\mathcal{N}_{i+1}$, $i \to i+1$ and the next step is B.1.

C. 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 $\mathcal{N}_0$.

D. (Optionally) The redundant constraints of (L, b) and (L_0, b_0) are identified and removed.

E. The supervisor of $\mathcal{N}_0$ is built according to the constraints $L\mu \geq b$ (Theorem 2.2).

4.5 Illustrative Examples

Example 1: Consider the repetitive Petri net of figure 2(a), where t_1 is unobservable. In the first iteration there are two minimal siphons: $\{p_1, p_3\}$ and $\{p_2, p_3\}$. Consider the siphon $\{p_1, p_3\}$. The marking constraint $\mu(p_1) + \mu(p_3) \geq 1$ is not admissible, so it is transformed to the following form (5): $2\mu(p_1) + \mu(p_3) \geq 1$. The control place C_1 is added according to this constraint, and the place invariant $\mu(C_1) = 2\mu(p_1) + \mu(p_3) - 1$ results. Similarly C_2 enforces $2\mu(p_2) + \mu(p_3) \geq 1$ on $\{p_2, p_3\}$ and $\mu(C_2) = 2\mu(p_2) + \mu(p_3) - 1$. The matrices L and b after the first iteration are:

$$L = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

In the second iteration there is a single new minimal siphon, $\{C_1, C_2\}$. The control place which would result by enforcing $\mu(C_1) + \mu(C_2) \geq 1$ is C_3 such that $C_3\bullet = \emptyset$. Therefore, according to the step B.2.(a) of the procedure, $\{C_1, C_2\}$ does not need control. $\mu(C_1) + \mu(C_2) \geq 1$ written in the form $l^T\mu \geq c$ is $2\mu(p_1) + 2\mu(p_2) + 2\mu(p_3) \geq 3$, and so

$$L_0 = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \qquad b_0 = \begin{bmatrix} 3 \end{bmatrix}$$

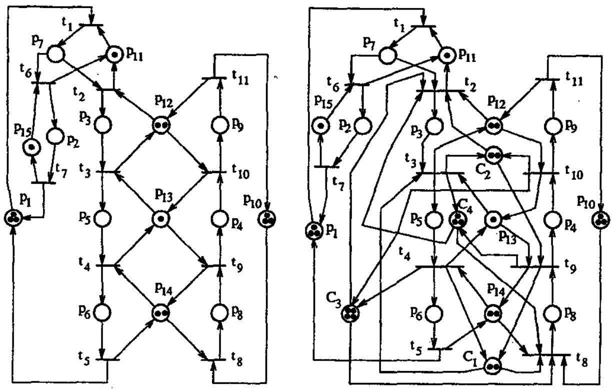

Figure 3: Target PN (left) and supervised PN (right).

The procedure terminates at the third iteration, since no new uncontrolled siphon is found. The supervised net is shown in Figure 2(b). By Theorem 5.1, liveness is enforced for all initial markings such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$, that is for $\mu_0 \in \{z \in \mathbb{N}^3 : (z(1) > 0 \wedge z(2) > 0) \vee (z(1) > 0 \wedge z(3) > 0) \vee (z(2) > 0 \wedge z(3) > 0) \vee (z(3) > 1)\}$. Moreover, by Theorem 5.2, the supervisor is maximally permissive.

Example 2: We apply our procedure to the Petri net of Fig. 3(c) in [2]. The Petri net and the supervised Petri net which results by applying our procedure are represented in Figure 3. The target Petri net is ordinary, but not with asymmetric choice. Before the first iteration, the AC-transformation is applied, and thus t_2 is split in t_2 and t_2', and a new place p_7' results.

At the first iteration the following minimal siphons are found: $S_1 = \{p_6, p_8, p_{14}\}$, $S_2 = \{p_2, p_{15}\}$, $S_3 = \{p_1, p_2, p_3, p_5, p_6, p_7, p_7'\}$, $S_4 = \{p_4, p_5, p_{13}\}$, $S_5 = \{p_4, p_6, p_{13}, p_{14}\}$, $S_6 = \{p_3, p_9, p_{12}\}$, $S_7 = \{p_5, p_9, p_{12}, p_{13}\}$, $S_8 = \{p_6, p_9, p_{12}, p_{13}, p_{14}\}$, $S_9 = \{p_7, p_{11}, p_7'\}$ and $S_{10} = \{p_4, p_8, p_9, p_{10}\}$. S_5, S_7 and S_8 generate the control places C_1, C_2 and C_3, while the others generate constraints in (L_0, b_0).

At the second iteration, the following new minimal siphons are found: $S_1 = \{p_5, p_8, C_1\}$, $S_2 = \{p_4, p_5, C_1, C_2\}$, $S_3 = \{p_4, p_6, p_{14}, C_1, C_2\}$, $S_4 = \{p_5, p_9, p_{12}, C_1, C_2\}$, $S_5 = \{p_6, p_9, p_{12}, p_{14}, C_1, C_2\}$, $S_6 = \{p_3, p_4, C_2\}$ and $S_7 = \{p_3, p_4, p_5, p_8, C_3\}$. The only uncontrolled siphon is S_2, which generates the fourth control place C_4.

At the third iteration there is only one new minimal siphon: $S = \{p_3, p_8, C_4\}$. S is not uncontrolled, and so no new control place is added. Therefore, no new siphons result and so the procedure terminates.

The liveness enforcing supervisor is defined by $L\mu \geq b$, where (L, b) contain the following (we let $\mu_i = \mu(p_i)$):

$$\mu_4 + \mu_6 + \mu_{13} + \mu_{14} \geq 1 \quad (6)$$

$$\mu_5 + \mu_9 + \mu_{12} + \mu_{13} \geq 1 \quad (7)$$

$$\mu_6 + \mu_9 + \mu_{12} + \mu_{13} + \mu_{14} \geq 1 \quad (8)$$

$$2\mu_4 + 2\mu_5 + \mu_6 + \mu_9 + \mu_{12} + 2\mu_{13} + \mu_{14} \geq 3 \quad (9)$$

The initial marking μ_0 must also satisfy the constraints of (L_0, b_0):

$$\mu_{0,6} + \mu_{0,8} + \mu_{0,14} \geq 1 \quad (10)$$

$$\mu_{0,2} + \mu_{0,15} \geq 1 \quad (11)$$

$$\mu_{0,1} + \mu_{0,2} + \mu_{0,3} + \mu_{0,5} + \mu_{0,6} + \mu_{0,7} \geq 1 \quad (12)$$

$$\mu_{0,4} + \mu_{0,5} + \mu_{0,13} \geq 1 \quad (13)$$

$$\mu_{0,3} + \mu_{0,9} + \mu_{0,12} \geq 1 \quad (14)$$

$$\mu_{0,7} + \mu_{0,11} \geq 1 \quad (15)$$

$$\mu_{0,4} + \mu_{0,8} + \mu_{0,9} + \mu_{0,10} \geq 1 \quad (16)$$

By Theorems 5.1 and 5.2, the supervisor enforces liveness and is maximally permissive. Comparing our result to [2], firing t_8, t_8, t_1, t_2, t_9, t_{10}, t_3, t_4, t_5, t_{11}, t_9, t_{10}, t_{11} from the marking of Figure 3 is allowed in our approach, but not allowed in [2] (see Fig. 6 in [2]). This emphasizes the fact that our supervisor is least restrictive. However this advantage may come at the price of increased computational complexity.

5 Theoretical Results

In this section we present our main theoretical results. We provide the proofs in the technical report [5].

5.1 Performance Results

Theorem 5.1 *Assume that the procedure terminates, and that it terminates at the step E. Let (L, b) and (L_0, b_0) denote the two sets of constraints generated by the procedure. The target net $\mathcal{N}_0$ supervised according to $L\mu \geq b$ is live for all initial markings μ_0 of $\mathcal{N}_0$ satisfying $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$.*

If a siphon control failure occurs at a step B.2.b, the procedure terminates at that step instead of step E. Note that such a failure is possible only for Petri nets which contain uncontrollable and/or unobservable transitions. The theorem also implies that the procedure will not terminate at step E if $\mathcal{N}_0$ is not repetitive, for liveness cannot be enforced in such a Petri net. However it can be easily verified whether a Petri net structure is repetitive. For instance the algorithm of [4] for the computation of the *maximal active subnet* may be used.

The next theorem states that the liveness enforcement supervisor provided by the procedure is least restrictive. The condition required by the theorem is true for all fully controllable and observable Petri nets and for some Petri nets with uncontrollable and unobservable transitions. Indeed, for fully controllable and observable Petri nets, (2) is always admissible, and therefore always in the form (5).

Theorem 5.2 *Assume that for all minimal active siphons S the procedure is able to find admissible constraints of the form (5) with all α_p positive integers. Then the liveness enforcement supervisor provided by the procedure is least restrictive.*

Theorems 5.1 and 5.2 imply that in the case of fully controllable and observable Petri nets, the procedure will not terminate unless the least restrictive liveness enforcing supervisor can be represented as a set of linear marking inequalities.

5.2 Modification for Guaranteed Termination

The procedure can be modified to guarantee termination. The modification we propose affects the performance of the supervisor. Thus Theorem 5.2 is no longer guaranteed to apply, but Theorem 5.1 still applies.

The modification of the procedure for termination is as follows. The siphon control method is modified. Let S be an uncontrolled siphon. Instead of enforcing $\sum_{p \in S} \mu(p) \geq 1$, the constraint $\sum_{p \in S \cap R} \mu(p) \geq 1$ is used, where R is the set of places which have not been generated by transiton splits. When uncontrollable and unobservable transitions are present, the latter form of the inequality is used for the transformation to an admissible constraint. Furthermore, we start the procedure with nonempty constraints (L_0, b_0). This requires an additional operation in step A: (L_0, b_0) are transformed according to the PT and AC transformations performed to obtain $\mathcal{N}_1$ from $\mathcal{N}_0$.

Theorem 5.3 *Let $\mathcal{N}$ 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. The modified liveness enforcement procedure terminates if started with initial constraints (L_0, b_0) which equal (L_i, b_i).*

Note that in the case of nonrepetitive Petri nets, because Theorem 5.1 still applies, after the procedure terminates there will be no initial marking to satisfy all constraints. Indeed, when no initial constraints are given, L, L_0, b and b_0 are nonnegative, and so there is always μ such that $L_0\mu \geq b_0$ and $L\mu \geq b$. This may no longer be the case when at the beginning of the procedure (L_0, b_0) is initialized to (L_i, b_i), since L_i and b_i may not be nonnegative.

The usage of the procedure modified for termination is outlined below. Note that this procedure modification is applicable to bounded Petri nets.

- Find a set of constrains $L_i\mu \geq b_i$ with bounded feasible set F such that for all initial markings μ_0 of $\mathcal{N}$ which are of interest: $\mathcal{R}(\mathcal{N}, \mu_0) \subseteq F$. Let $\mathcal{M}_I$ be the set of initial markings of interest.
- Use the modified procedure with 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.

6 Conclusion

This paper presents a liveness enforcement procedure. The supervisors generated enforce liveness and, in the case of Petri nets with controllable and observable transitions, are least restrictive. The superviors are defined as a set of linear marking inequalities, and are independent of the initial marking.

References

[1] K. Barkaoui and J. F. Pradat-Peyre. On liveness and controlled siphons in Petri nets. In *LNCS; Proc. 17th International Conference in Application and Theory of Petri Nets*, vol. 1091, pp 57–72. Springer-Verlag, June 1996.

[2] J. Ezpeleta, J. M. Colom, and J. Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 11(2):173–184, 1995.

[3] K.X. He and M.D. Lemmon. On the transformation of maximally permissive marking-based supervisors into monitor supervisors. In *Proc. of the 39'th IEEE Conf. on Decision and Control*, pp 2657-2662, December 2000.

[4] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. Automated synthesis of deadlock prevention supervisors using Petri nets. Technical report of the isis group, isis-2000-003, University of Notre Dame, May 2000.

[5] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. Automated synthesis of liveness enforcement supervisors using Petri nets. Technical report of the isis group, isis-2000-004, University of Notre Dame, September 2000.

[6] M. V. Iordache, J. O. Moody, and P. J. Antsaklis. A method for the synthesis of deadlock prevention controllers in systems modeled by Petri nets. In *Proceedings of the American Control Conference*, pp 3167-3171, June 2000.

[7] K. Lautenbach and H. Ridder. The linear algebra of deadlock avoidance – a petri net approach. Technical report, Univ. of Koblenz, Institute for Comp. Sci., 1996.

[8] John O. Moody and Panos J. Antsaklis. *Supervisory Control of Discrete Event Systems Using Petri Nets.* Kluwer Academic Publishers, 1998.

[9] T. Murata. Petri nets: Properties, analysis and applications. In *Proc. of the IEEE*, pages 541–580, April 1989.

[10] S. R. Sreenivas. An application of independent, increasing, free-choice Petri nets to the synthesis of policies that enforce liveness in arbitrary Petri nets. *Automatica*, 44(12):1613–1615, December 1998.

[11] S. R. Sreenivas. On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement. *IEEE Transactions on Automatic Control*, 44(1):173–177, January 1999.

[12] E. Yamalidou, J. O. Moody, P. J. Antsaklis, and M. D. Lemmon. Feedback control of petri nets based on place invariants. *Automatica*, 32(1):15–28, January 1996.