A DEADLOCK PREVENTION APPROACH FOR A CLASS OF TIMED PETRI NETS USING ELEMENTARY SIPHONS

Jinwei Guo and Zhiwu Li

ABSTRACT

To solve the problem of deadlock prevention for timed Petri nets, an effective deadlock prevention policy based on elementary siphons is proposed in this paper. Without enumerating reachable markings, deadlock prevention is achieved by adding monitors for elementary siphons, increasing control depth variables when necessary, and removing implicit, liveness-restricted and redundant control places. The final supervisor is live. First, a timed Petri net is stretched into a stretched Petri net (SPN). Unchanging the system performance, each transition in the SPN has a unit delay time. Then the siphon-control-based approach is applied. Monitors computed according to the marking constraints are added to the SPN model to ensure all strict minimal siphons in the net invariant-controlled. A liveness-enforcing supervisor with simple structure can be obtained by reverting the SPN into a TdPN.

Key Words: Flexible manufacturing systems, timed Petri nets, deadlock prevention, elementary siphon.

I. INTRODUCTION

Petri nets are a kind of graphical and mathematical tool which is widely used in modeling discrete event systems (DES). System designers interact with users by designing a Petri net structure according to some particular objectives. They verify the properties of objects by means of Petri nets based on analysis methods. The Petri net model of a system is a directed bipartite graph which includes two types of nodes called places, represented by circles, and transitions, represented by bars. A place usually models a condition and a transition usually models an event. A system state is denoted by a marking of Petri net, which is the number of tokens in the place. The Petri net model can be also expressed in mathematical form, i.e., incidence matrix. This representation is proper to analyze structure and behavior of a system. Reachability trees are also used in system behavior analysis [1, 2].

When modeling a system from a logical point of view, Petri nets do not consider the notion of time. However, for performance analysis and real-time control, the concept of time should be synthesized into Petri nets, which forms Timed Petri nets (TdPNs) [2, 3]. In general, timed Petri net can be classified into three types, implying that temporal constraints can be associated with places, transitions, or arcs. In this research, we consider timed-transition Petri nets only. For this, in general, a system that is modeled as a timed-place Petri net can alternatively be modeled as a timed-transition one. Furthermore a timed-transition Petri net is closer to real-time systems. Timed Petri nets enhance the modeling capacity of Petri net [4]
such that the modeling capacity of TdPNs is equal to that of Turing machines. At the same time, however, TdPNs alter the basic structures and behaviors of Petri nets and make system analysis more difficult. Wang et al. [5] proposed a technique based on a concept called clock-stamped state class (CS-class) which not only groups system states into compact representation of state classes, but also records the time, relative to the beginning of system execution. Whereas the method is based on the reachability tree analysis of the system, it is appropriate only for the case in which the end-to-end time delay in task execution is required. Salum [6] proposes a method of modeling temporal constraints through the basic Petri net semantics, without changing any enabling or firing rules as in TdPNs. Salum considers time as resources and states of the system. This formalism makes it natural to incorporate time into basic Petri net semantics. Thus a variety of structures such as inhibitor arcs and test arcs are naturally introduced, which makes a natural connection between the time and logic relation. Methods based on reachability tree analysis can be applied to TdPNs. The technique proposed by Salum [7] is also apt to end-to-end real-time event and problem of event delay. Aybar [8] proposed the notion called stretched petri net (SPN) which has only unit delay for each transition. With the method based on the reachability tree, TdPN analysis is simplified. In [8], a deadlock avoidance policy is proposed.

In [2, 6, 5, 7–10], researchers transform TdPNs into PNs or similar forms, and analyze TdPNs through the existing Petri net analysis methods. However the shortcomings are that all of them utilize the transformed reachability tree to investigate the system properties. It is well known that the scale of system reachability tree exponentially grows with the size of a Petri net, which makes it difficult to analyze the properties of systems with a large number of transitions and places. The authors of [8, 7, 10] develop online deadlock detection or avoidance strategies, which are not convenient to analyze the system in which real-time is emphasized especially.

In this paper, the concept of SPN is employed, and a deadlock prevention policy for a class of TdPNs, called TdS$^3$PR, is proposed, which overcomes the disadvantages mentioned above. First, a TdPN is stretched into an SPN. Thus, time factor can be ignored. For each elementary siphon, a monitor is added such that it is invariant-controlled. After testing the controllability of dependent siphons and enlarging the depth variables when necessary, each siphon is successfully controlled and no emptiable control-induced siphons can be produced. Then implicit and liveness-restricted places are removed. Finally, the SPN can be reverted into a TdPN. After the removal of redundant control places, the final TdPN with simple structure is a liveness-enforcing supervisor.

The rest of this paper is organized as follows. Section II reviews basic definitions of timed and stretched Petri nets. It also gives the definition of elementary siphons and the concept of TdS$^3$PR nets. Section III proposes a deadlock prevention policy for TdS$^3$PR. Section IV introduces an FMS example to illustrate the application of the proposed policy. Conclusions and further research topics are presented in Section V.

II. PRELIMINARIES

2.1 Petri nets

A Petri net [1] is a 4-tuple $N=(P, T, F, W)$ where $P$ and $T$ are finite, non-empty and disjoint sets. $P$ is the set of places and $T$ is the set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called the flow relation or the set of directed arcs. $W : F \rightarrow \mathbb{N}$ is a mapping that assigns a weight to any arc, where $\mathbb{N} = \{0, 1, 2, \ldots \}$. The incidence matrix $[N]$ of net $N$ is a $|P| \times |T|$ integer matrix and $[N](p, t) = post(t, p) - pre(p, t)$. The preset of a node $x \in P \cup T$ is defined as $x = \{ y \in P \cup T \mid (y, x) \in F \}$. The postset of a node $x \in P \cup T$ is defined as $x = \{ y \in P \cup T \mid (x, y) \in F \}$. This notation can be extended to a set of nodes as follows: given $X \subseteq P \cup T$, $X = \bigcup_{x \in X} x$ and $X = \bigcup_{x \in X} x$. A marking is a mapping $M : P \rightarrow \mathbb{N}$.

The pair $(N, M_0)$ is called a marked Petri net or a net system. The set of markings reachable from $M$ in $N$ is denoted as $R(N, M)$. $(N, M_0)$ is bounded iff $\exists k \in \mathbb{N}, \forall M \in R(N, M_0), \forall p \in P, M(p) \leq k$ holds. A transition $t \in T$ is enabled under $M$, denoted by $M[t]$, iff $\forall p \in \text{post}(t), M(p) \geq 1$. A transition $t \in T$ is live under $M_0$ iff $\forall M \in R(N, M_0), \exists M' \in R(N, M), M'[t]$ holds. $(N, M_0)$ is deadlock-free iff $\forall M \in R(N, M_0), \exists t \in T, M[t]$ holds. $(N, M_0)$ is live iff $\forall t \in T, t$ is live under $M_0$.

P-vector $I$ is a P-invariant (place invariant) iff $I \neq 0$ and $I^T[N] = 0^T$. P-invariant $I$ is said to be a P-semiflow if no element of $I$ is negative. $\|I\| = \{ p \in P \mid I(p) \neq 0 \}$ is called the support of $I$. An invariant is minimal when its support is not a strict superset of the support of any other, and the greatest common divisor of its elements is one.
A set of places \( S \subseteq P \) is a siphon iff \( \bullet S \subseteq S \bullet \). A siphon is minimal iff there is no siphon contained in \( S \) as a proper subset. A minimal siphon is said to be a strict minimal siphon (SMS) if it does not contain the support \( ||I|| \) of a \( P \)-invariant \( I \). Let \( N \) be a net with \( k \) SMS, \( S_1, S_2, \ldots, S_k \). We use \( \Pi=\{S_1, S_2, \ldots, S_k\} \) to denote the set of SMS in \( N \).

### 2.2 Timed PN and SPN

Unless otherwise stated, we consider only ordinary and bounded Petri nets in this paper. Without considering the behavior duration, basic Petri nets are not enough to investigate system performance. In order to model systems better, some definitions about extended Petri nets with time property are given below, and also be found in [8]. Let \( \mathbb{N}^+ \) be the set of positive integers.

**Definition 1.** A timed Petri net is defined as a 4-tuple \( N=\langle P, T, F, D \rangle \), where \( \langle P, T, F \rangle \) is a Petri net and \( D \) denotes the set of delay time \( d_t \), where \( d_t \in \mathbb{N}^+ \) is the delay time of transition \( t \in T \). A net system \( \langle N, M_0 \rangle \) can also be denoted as \( \langle P, T, F, M_0, D \rangle \).

**Definition 2.** The incidence matrix of TdPN \( \langle N, M_0 \rangle=\langle P, T, F, M_0, D \rangle \) is defined as a \( |P|\times|T| \) integer matrix, \( [N]=\text{Post}−\text{Pre} \), where \( \text{Pre}: P\times T \rightarrow \mathbb{N} \) is the input matrix and \( \text{Post}: T\times P \rightarrow \mathbb{N} \) is the output matrix. That is, if \((p,t)\in F\) then \( \text{pre}(p,t)=1 \), else \( \text{pre}(p,t)=0 \); if \((t,p)\in F\) then \( \text{post}(p,t)=1 \), else \( \text{post}(p,t)=0 \).

As in [8], we also assume that the time can be discretized using an appropriate sampling period. Thus, we represent the time variable by an integer \( k \) and let the initial time be \( k=0 \). The state of TdPN at time \( k \) is represented by \( S(k)=\{M(k),Q(k)\} \), where \( M(k):P \rightarrow \mathbb{N} \) is the marking vector of time \( k \) and \( Q(k):\{(t,k−\kappa_f(t))|k−d_t<\kappa_f(t)<k\} \), where \( \kappa_f(t) \) denotes the firing time of transition \( t \in T \). A transition \( t \in T \) is enabled at marking \( M \) iff \( \forall p\in \bullet t \), \( M(p)\geq \text{pre}(p,t) \).

To simplify the representation of a TdPN’s states, a method named “stretching” is introduced [8], which converts an original TdPN (OPN) into another form, called a stretched Petri net (SPN). For an SPN, each transition has a unit delay time. It is defined as follows:

**Definition 3.** For a TdPN, \( T \) is divided into two disjoint subsets \( T_1=\{t\in T|d_t=1\} \) and \( T_2=\{t\in T|d_t\geq2\} \). Then \( T=\bigcup T_2 \).

For any \( t \in T_2 \), we process as below:

1. Define \( d_t−1 \) new transitions \( t_{d_t−1}^{1}, t_{d_t−1}^{2}, \ldots, t_{d_t−1}^{d_t−1} \) and \( d_t−1 \) new places \( p_{d_t−1}^{1}, p_{d_t−1}^{2}, \ldots, p_{d_t−1}^{d_t−1} \). Let \( \sigma_T(t)=\{t_1^{d_t−1}, t_2^{d_t−1}, \ldots, t_{d_t−1}^{d_t−1}\} \) and \( \sigma_P(t)=\{p_1^{d_t−1}, p_2^{d_t−1}, \ldots, p_{d_t−1}^{d_t−1}\} \). Then \( \forall t' \in \sigma_T, t' \in T, \) once enabled, it fires immediately.
2. Keep all the input arcs of \( t \) unchanged, but disjoin all the output arcs of \( t \). Then attach \( t \) to \( p_1^{d_t−1} \) and attach the originating end of output arcs to transition \( t_{d_t−1}^{d_t} \).
3. Add arcs: \( t \rightarrow p_{d_t−1}^{1}, p_1^{d_t−1} \rightarrow t_{d_t−1}^{1}, t_{d_t−1}^{d_t−1} \rightarrow p_{d_t−1}^{d_t−1} \). All of the new arcs have unit weight and all the new introduced transitions have a unit delay time.
4. Reassign the delay of \( t \) as unit.

Then a new TdPN, namely SPN, which has more transitions and places is achieved. However, the delay time of each transition is unit in SPN.

For example, Fig. 1(a) is a TdPN and (b) is its corresponding SPN. In Fig. 1(a), \( t_1 \) and \( t_2 \) have two units delay time, respectively. In Fig. 1(b) each transition has a unit delay time.

**Definition 4.** A SPN is a 4-tuple \( (N_s, M_{s0})=(P_s, T_s, F_s, M_{s0}) \), where \( P_s=P \cup \bar{P} \) is the set of places, \( T_s=T \cup \bar{T} \) is the set of transitions, \( T_0=\{t|t \in T_2\} \) and \( F_s \subseteq (P_s \times T_s) \cup (T_s \times P_s) \). Incidence matrix \( [N_s]=\text{Post}−\text{Pre} \), where \( \text{Post}_s: P_s \times T_s \rightarrow \mathbb{N} \) is input matrix; \( \text{Post}_s: T_s \times P_s \rightarrow \mathbb{N} \) is output matrix; and \( M_{s0}:P_s \rightarrow \mathbb{N} \) is the initial markings. \( M_{s0}=U^T M_0 \), where \( U \) is the transformation matrix defined as: \( U \cdot P \rightarrow [0,1] \) and

\[
U(p_a, p_b) = \begin{cases} 
1 & \text{if } p_a = p_b \\
0 & \text{otherwise}
\end{cases}
\]

The input and output matrices are defined as follows:

\[
\text{pre}(p,t) = \begin{cases} 
\text{pre}(p,t) & \text{if } p \in P \text{ and } t \in T \\
1 & \text{if } p = p^1_{i} \text{ and } t = t^1_{i} \\
& \text{for some } i \in T_2 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{post}(p,t) = \begin{cases} 
\text{post}(p,t) & \text{if } p \in P \text{ and } t \in T_1 \\
1 & \text{if } p = p^1_{i} \text{ and } t = t^1_{i} \\
& \text{for some } i \in T_2 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{pre}(p,t) = \begin{cases} 
\text{pre}(p,t) & \text{if } p \in P \text{ and } t \in T_1 \\
1 & \text{if } p = p^1_{i} \text{ and } t = t^1_{i} \\
& \text{for some } i \in T_2 \text{ with } d_i \geq 3 \\
0 & \text{otherwise}
\end{cases}
\]
It can be seen that, the state of an SPN at time $k$ is determined completely by the marking vector at that moment. Moreover, this definition can be used to determine the state, $S(k)=\{M(k), Q(k)\}$, of the original PN, i.e., $M(k)=UM_s(k)$ and $Q(k)=\{(t, \kappa)\}|M_s(p_i^k, t)=1$.

For example in Fig. 1,

$$U = \begin{pmatrix}
 p_1 & p_2 & p_3 & p_1^{(1)} & p_1^{(2)} \\
 p_1 & 1 & 0 & 0 & 0 \\
 p_2 & 0 & 1 & 0 & 0 \\
 p_3 & 0 & 0 & 1 & 0 \\
\end{pmatrix}. $$

We presume the time $k=3$. Then, for Fig. 1a, we have $M(k)=[0, 0, 0]^T$ and $Q(k)=\{(t_2, 1)\}$. For Fig. 1b, we have $M_s(k)=[0, 0, 0, 1]^T$, $Q(k)=\{(t_2, 1)\}$, and $M(k)=UM_s(k)$. From the SPN, we can clearly see the distribution of tokens at any moment in the net.

With the transformation from a TdPN (original PN) to an SPN, the properties of system do not change. However, owing to the structure simplicity, some available methods can be used, e.g., deadlock avoidance and prevention approaches, to achieve deadlock control purpose.

### 2.3 Elementary and dependent siphons of petri nets

This subsection presents the concepts of elementary and dependent siphons in a Petri net. They can also be found in literatures [11–14].

**Definition 5.** Let $N=(P, T, F)$ be a Petri net and $S \subseteq P$ be a siphon of $N$. $P$-vector $\lambda_S$ is called the characteristic $P$-vector of $S$ if $\forall p \in S$: $\lambda_S(p)=1$; otherwise $\lambda_S(p)=0$.

**Definition 6.** Let $N=(P, T, F)$ be a Petri net, $S \subseteq P$ be a siphon of $N$, and $\lambda_S$ be the characteristic $P$-vector of $S$. $\eta_S$ is called the characteristic $T$-vector of $S$ if $\eta_S^T = \lambda_S^T |N|$.

**Theorem 1.** Let $S$ be a siphon of net $N=(P, T, F)$, and $\eta_S$ be its characteristic $T$-vector. Then

1. The number of tokens in $S$ will increase if $t \in\{t' \in T | \eta_S(t')>0\}$ is fired.
2. The number of tokens in $S$ will maintain if $t \in\{t' \in T | \eta_S(t')=0\}$ is fired.
3. The number of tokens in $S$ will decrease if $t \in\{t' \in T | \eta_S(t')<0\}$ is fired.

**Definition 7.** Let $(N, M_0)$ have $k$ strict minimal siphons (SMS) $S_1, S_2, \ldots, S_k$, where $N=(P, T, F)$, $|P|=m$, and $|T|=n, m, n, k \in \mathbb{N}$. Let $\lambda_S$ be the characteristic $P(T)$-vector of SMS $S_i$, $i \in\{1, 2, \ldots, k\}$. We define $[\lambda]_{N \times m}=[\lambda_{S_1}, \lambda_{S_2}, \ldots, \lambda_{S_k}]^T$ and $[\eta]_{k \times n}=[\eta_{S_1}, \eta_{S_2}, \ldots, \eta_{S_k}]^T$. $[\lambda](\eta)$ is called the characteristic $P(T)$-vector matrix of the SMSs in $N$.

**Definition 8.** Let $(N, M_0)$ have $k$ strict minimal siphons $S_1, S_2, \ldots, S_k$. $\{S_2, S_3, \ldots, S_k\}$ is called a set of elementary siphons in $N$ if $\eta_{S_1}^T \eta_{S_2} \eta_{S_3} \cdots \eta_{S_k}^T$ are the linearly independent maximal set of matrix $[\eta]$, where $\{x, \beta, \gamma\} \subseteq\{1, 2, \ldots, k\}$.

We use $\Pi$ to denote the set of SMSs in a net $N$ and $\Pi_E$ denote the set of elementary siphons.

**Definition 9.** Let $\Pi_E = \{S_2, S_3, \ldots, S_k\}$ be a set of elementary siphons in $N$. $S \in \Pi \setminus \Pi_E$ is called a strongly dependent siphon if $\eta_S = \sum_{i \in \{x, \beta, \gamma\}} a_i \eta_{S_i}$, where $a_i \geq 0$.

**Definition 10.** Let $\Pi_E = \{S_2, S_3, \ldots, S_k\}$ be a set of elementary siphons in $N$. $S \in \Pi \setminus \Pi_E$ is called a weakly dependent siphon if $\exists$ nonempty $S^1, S^2 \subseteq \Pi_E$ such that $S^1 \cap S^2 = \emptyset$ and $\eta_S = \sum_{i \in S^1} a_i \eta_{S_i} - \sum_{i \in S^2} a_j \eta_{S_j}$, where $a_i > 0$ and $a_j > 0$.

The strongly and weakly dependent siphons are called dependent ones by a joint name. That is if $\eta_S$ can be linearly represented by elementary siphons’ characteristic $T$-vectors $\eta_{S_1}, \eta_{S_2}, \ldots, \eta_{S_k}$ with nonzero coefficients, we may say that $S_2, S_3, \ldots, S_k$ are the elementary siphons of $S$. Let $\Pi_D$ be the set of dependent siphons within the scope of $\Pi$. Obviously, we have $\Pi = \Pi_E \cup \Pi_D$.

**Theorem 2 ([12]).** Let $N_{ES}$ be the number of elementary siphons in net $N=(P, T, F)$. Then we have $N_{ES} \leq r(|N|) \leq \min\{(|P|, |T|)\}$, where $r(|N|)$ is the rank of $|N|$.
choose tary siphons and
position, there are three strict minimal siphons that
for deadlock control purpose: \( S_1 = \{p_4, p_6, p_{13}, p_{14}\} \), \( S_2 = \{p_5, p_9, p_{12}, p_{13}\} \), and \( S_3 = \{p_6, p_9, p_{12}, p_{13}\} \). We have \( \lambda S_1 = p_4 + p_6 + p_{13} + p_{14} \), \( \lambda S_2 = p_5 + p_9 + p_{12} + p_{13} \), \( \lambda S_3 = p_6 + p_9 + p_{12} + p_{13} + p_{14} \), \( \eta S_1 = -t_3 + t_4 - t_5 + t_9 \), \( \eta S_2 = -t_2 + t_3 - t_9 + t_{10} \), and \( \eta S_3 = -t_2 + t_4 - t_8 + t_{10} \). It is easy to verify that \( \eta S_1 = \eta S_2 + \eta S_3 \) which means that there are two elementary
siphons and \( S_3 \) is strongly dependent. Also if we
choose \( S_2 \) and \( S_3 \) to be elementary siphons, \( S_1 \) is weakly
dependent. Likewise, in the Fig. 2(b), there are three
strict minimal siphons \( S_1 = \{p_4, p_6, p_{13}, p_{14}, p_1^{13}, p_1^{15}, p_1^{11}\} \), \( S_2 = \{p_5, p_9, p_{12}, p_{13}, p_3^{11}, p_1^{11}, p_1^{11}, p_1^{11}\} \), and
\( S_3 = \{p_6, p_9, p_{12}, p_{13}, p_{14}, p_4^{11}\} \) which are all elementary, implying that there is
no dependent siphon in the net model.

As is known, a siphon is a set of places whose
tokens can leak out and never return, making it dead.
If no siphon becomes empty, the net is deadlock-free.
For an \( S^3PR \), we can use the method mentioned in [15]
to make it live without bringing new strict minimal
siphons. However, that method makes the net super-
visor size too large since many control places and arcs
are introduced. Also it restricts the number of reachable
states of net system. The results in [11] show that all
SMRs can be controlled by making the elementary ones
controlled by choosing a proper control depth variable
for each dependent siphon. The latter control policy is
of linear complexity, from a structural point of view,
with respect to the size of a plant net. However, the
former is exponential. Next section proposes a deadlock

Fig. 2. A TdPN and its SPN version. (a) A plant net model \((N, M_0)\), (b) A SPN net \((N_s, M_{s0})\).
prevention policy for timed $S^3PR$ (TdS$^3PR$) based on the elementary siphons.

**Definition 11** ([16]). Let $(N, M_0)$ be a net system and $S$ be a siphon. $S$ is called a well-behaved siphon if $\forall M \in R(N, M_0), M(S) > 0$.

Siphon $S$ is well-behaved if $\min\{M(S)\} M = M_0 + [N], M \geq 0, Y \geq 0 > 0$.

2.4 Implicit places and liveness-restricted places

**Definition 12.** Let $(N, M)$ be a Petri net. The set of firing sequences from $M$ is called the language of $N$, denoted by $L(N, M)$.

Implicit places (IPs) are a kind of places with the property that their addition to or removal from a net system does not change its behavior, i.e., an IP is independent from transition enabling and it can be removed [17].

**Definition 13** ([18]). Let $(N, M_0)$ be a net system and $(N', M'_0)$ the net system resulting from removing place $p$ from $N$. Place $p$ is a sequential implicit place (SIP) iff $L(N, M_0) = L(N', M'_0)$. If, additionally, $R(N, M_0) = R(N', M'_0)$, then $m(p)$ is marking dependent and $p$ is said to be a marking-dependent implicit place.

It can be reached from the definition that the elimination of an implicit place preserves deadlock-freeness, liveness, or marking mutual exclusion. If it is marking-dependent, then it additionally preserves boundedness or reversibility. Note that the removal of tokens from an IP can make the place not implicit any longer. That is to say, the implicit property is interrelated to the markings of the net. And an IP remains implicit for all initial markings greater than a certain minimum.

**Theorem 3** ([17, 18]). Let $(N, M_0)$ be a net system. A place $p \in P$ with initial marking $M_0(p)$ is implicit if $M_0(p) \geq \max\{0, z\}$, where $z$ is the optimal value of the following LPP

$$z = \min\{y \cdot M_0 + \mu\}$$

s.t.

$$y \cdot [N] \leq [N](p, T)$$

$$y \cdot \text{Pre}(p, T) + \mu \geq \text{Pre}(p, T), \ \forall t \in P^*$$

$$y \geq 0, \ y(p) = 0.$$

The conditions of Theorem 3 are sufficient but not necessary, which means that $p$ can still be implicit if $M_0(p) < \max\{0, z\}$.

**Definition 14.** Let $(N, M_0)$ be a live net system and $(N', M'_0)$ the net system resulting from removing place $p$ and its related arcs from $N$. Place $p$ is a liveness-restricted place if $(N', M'_0)$ is live.

It follows that removing a liveness-restricted place can change the behavior of the net system but preserve its liveness property. In the usual opinion, the control places are used to restrict the behavior of net. Therefore, removing the liveness restricted places among the control places can theoretically increase the reachable states of a live net.

**Theorem 4** ([19]). Let $(N, M_0)$ be a net system. Then $N$ does not contain the emptyable siphon if $G^{MIP}(M_0) = |P|$, where $G^{MIP}(M_0) = \min\{\sum_{p \in P} v_p s.t.\}$

$$z_p \geq \sum_{p \in P^*} v_p - |t| + 1, \ \forall t \in T$$

$$v_p \geq z_p, \ \forall (t, p) \in F$$

$$v_p, z_p \in \{0, 1\}$$

$$v_p \geq M(p)/\psi(p), \ \forall p \in P$$

$$\psi(p) = \max\{M(p) M = M_0 + [N], Y, M \geq 0, Y \geq 0\}$$

$$M = M_0 + [N]. Y, M \geq 0, Y \geq 0.$$

According to Theorem 4, for a live controlled net system $(N_1, M_1)$ with the set of places $P_1$, we have $G^{MIP}(M_1) = |P_1|$. Let $(N_2, M_2)$ be the net removing $p$ from $N_1$. Then $N_2$ does not contain emptyable siphons if $G^{MIP}(M_2) = |P_1| - 1$.

2.5 $S^3PR$ and Td$S^3PR$

The proposed deadlock prevention policy targets $S^3PR$ [15]. This section introduces $S^3PR$ and Td$S^3PR$ net models. The latter plays a crucial role in establishing the deadlock prevention policy.

**Definition 15.** A Simple Sequential Process ($S^2P$) net is a strongly connected state machine $N = (P \cup \{p^0\}, T, F)$ with exactly one initially marked place $p^0$ (idle place) such that each circuit of the net contains $p^0$. The other places are called operation places.

**Definition 16.** A Simple Sequential Process with Resources ($S^3PR$) is a Petri net $N = (P \cup \{p^0\}, T, F)$ such that: 1) The subset generated by $X = P \cup \{p^0\}$ is an $S^2P$; 2) $P_R \neq \emptyset$ and $(P \cup \{p^0\}) \cap P_R = \emptyset$; 3) $\forall p \in P, \ \forall t \in P^*, \ \forall t \in P^*, t \cap P_R = t^* \cap P_R = \{r_p\}$; 4) The two following statements are verified: a) $\forall r \in P_R, \ r \cap P = r^* \cap P \neq \emptyset; b) \forall r \in P_R, \ r \cap r^* = \emptyset$; 5) $(p^0) \cap P_R = (p^0)^* \cap P_R = \emptyset$.

**Definition 17.** A System of $S^3PR$, $S^3PR$, is defined recursively as follows: 1) An $S^3PR$ is an $S^3PR$; 2) Let $N_i = (P_i \cup P^0_i \cup P_R, T_i, F_i), i \in \{1, 2\}$, be two $S^3PR$ such that $(P_1 \cup P^0_1) \cap (P_2 \cup P^0_2) = \emptyset, P_{R1} \cap P_{R2} = P_{C} \neq \emptyset$, and

© 2010 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society
Definition 18. Let \( N = \bigcup_{i=1}^{k} N_i = (P \cup P^0 \cup P_T, T, F) \) be an S3R and \( N \) be a strict minimal siphon in \( N \), where \( S = S_P \cup S_T, S_T = S \cap P_T, T, T \) is the set of transitions, \( F = T_1 \cup T_2 \), and \( F = T_1 \cup T_2 \) is also an S3R.

Definition 19. A TdS3PR is a timed Petri net \((N, M_0) = (P \cup P^0 \cup P_T, T, F, M_0, D)\), where \((N', M_0) = (P \cup P^0 \cup P_T, T, F, M_0)\) is a marked S3PR. \( D \) represents the set of delays \( d_i \), where \( d_i \in \mathbb{N}^+ \) is the delay time of \( d_i \).

Definition 20. An STdS3PR is the stretched TdS3PR \((N_s, M_s) = (P_s \cup P^0_s \cup P_T, T_s, F_s, M_s(0), D_s)\), where \( P_s = P \cup P_T \) is the set of operation places, \( P_s = \bigcup_{i \in T_s} \sigma_P(t) \), \( T_s = T \cup T_s \) is the set of transitions, \( T_s = \bigcup_{t \in T_s} \sigma_T(t) \), and \( F_s \subseteq ((P_s \cup P^0_s \cup P_T) \times T_s) \). Incident matrix \([N_s] = \text{Post}_{s} - \text{Pre}_{s}\), where \([N_s] = \text{Post}_{s} - \text{Pre}_{s}\) is the input matrix and \([N_s] = \text{Post}_{s} - \text{Pre}_{s}\) is the output matrix. \( M_0: (P_s \cup P^0_s \cup P_T) \to \mathbb{N} \) is the initial markings with \( M_0 = U^T M_0 \). \( D_s \), a \(|T_s| \times 1 \) unit column vector, is called the delay vector.

For example, Fig. 2(a) is a TdS3PR net model where \( t(k) \) means that the delay time of transition \( t \) is \( k \) units. If the delay time is one unit, it is not shown. The Fig. 2(b) corresponds to the STdS3PR net model where each transition has a unit delay time.

III. DEADLOCK PREVENTION POLICY OF TdS3PR

A TdPN with deterministic delay for each transition can be stretched. For an SPN, since each transition has a unit delay, the influence which time factor exerts on the reachability tree can be ignored. However, the addition of monitors should be realized on the original PN model. This means that a controlled SPN should be reverted into a PN and the resultant PN is also controlled.

Definition 21. Let \((N_s, M_s) = \bigcup_{i=1}^{k} N_i = (P_s \cup P^0_s \cup P_T, T_s, F_s, M_s(0), D_s)\) be a marked STdS3PR net system and \( S \) be a strict minimal siphon, where \( S = S_P \cup S_T, S_T = S \cap P_T, T, T \) is the support of a \( P \)-semiflow. (2) \([S]\) is called the complementary set of siphon \( S \). (3) \( \forall p \in \mathbb{S}, [S] \), represents the multi-set of places that do not belong to siphon \( S \) but use the resources of \( S \).

Theorem 5. Let \((N_s, M_s)\). \( N_s = \bigcup_{i=1}^{k} N_i = (P_s \cup P^0_s \cup P_T, T_s, F_s, D_s)\), be a marked STdS3PR net system and \( S \) be a weakly dependent siphon of \( N_s \). Let \( S_1, S_2, \ldots, S_n, S_{n+1}, S_{n+2}, \ldots, S_m \) be the elementary siphons of \( S \) with \( \eta_S = \sum_{i=1}^{n} a_i \eta_{S_i} - \sum_{j=n+1}^{m} a_j \eta_{S_j} \). \( S \) is controlled if \( M_{s0}(S) > \sum_{i=1}^{n} a_i (M_{s0}(S_i) - M_{s0}(S_j)) \).

Theorem 6. Let \((N_s, M_s)\). \( N_s = \bigcup_{i=1}^{k} N_i = (P_s \cup P^0_s \cup P_T, T_s, F_s, D_s)\), be a marked STdS3PR net system and \( S \) be a strongly dependent siphon of \( N_s \). Let \( S_1, S_2, \ldots, S_n \) be the elementary siphons of \( S \) with \( \eta_S = \sum_{i=1}^{n} a_i \eta_{S_i} \). \( S \) is controlled if \( M_{s0}(S) > \sum_{i=1}^{n} a_i (M_{s0}(S_i) - M_{s0}(S_j)) \).

Theorem 7. Let \( S \) be weakly dependent in \( N_s \) with \( \eta_S = \sum_{i=1}^{n} a_i \eta_{S_i} - \sum_{j=n+1}^{m} a_j \eta_{S_j} \). \( S \) is the elementary siphons of \( S \), where \( i \in \{1, 2, \ldots, n, n+1, \ldots, m\} \). Let \( D_1 = \min \{ \sum_{i=1}^{n} a_i M_s(S_i) \} \) and \( D_2 = \max \{ \sum_{j=n+1}^{m} a_j M_s(S_j) \} \) and \( D_1 = \min \{ \sum_{i=1}^{n} a_i M_s(S_i) \} \) and \( D_2 = \max \{ \sum_{j=n+1}^{m} a_j M_s(S_j) \} \).
Theorem 8. Let $S$ be strongly dependent in $N_i$ with $\eta_S = \sum_{i=1}^{n} a_i \eta_{S_i}$, $S_i$ being the elementary siphons of $S$, where $i \in \{1, 2, \ldots, n\}$. $S$ is controlled if $M_{i0}(S) > \sum_{i=1}^{n} a_i M_{i0}(S_i) - D_1$.

Obviously Theorem 7(8) is stronger than Theorem 5(6). For practical purposes, we can first use Theorem 5(6) to verify the controllability of a dependent siphon $S$. If $S$ is not controlled due to Theorem 5(6), Theorem 7(8) can be then employed. These theorems can be similarly proved as in [21].

Lemma 1. Let $(N_i, M_{i0})$ with $N_i = \bigcup_{i=1}^{k} N_{i\lambda} = (P_i \cup P_{s0} \cup P_{s2} \cup P_{s3})$, $S = S_{p} \cup S_{R}$ be a strict minimal siphon, where $S_{p} = S_{p1} \cap S_{p2} \cap S_{p3} \cup S_{R}$. Let $I_{S}$ be a $P$-vector of $N_i$ with $\{p|I_{S}(p) > 0\} \subseteq S$. Add a monitor $V_S$ to $(N_i, M_{i0})$ such that $I = (I_{S} + [1])^T$ is a $P$-invariant of the new net system $(N'_i, M'_{i0})$, where $\forall p \in P_{s} \cup P_{s2} \cup P_{s3}, M'_{i0}(p) = M_{i0}(p)$ and $M'_{i0}(V_S) = I_{S}^T M_{i0} - \xi_S$. Then, $S$ is invariant-controlled if $\xi_S > 0$.

Let $(N_i, M_{i0})$ be a STd3PR net system. Next we will develop two algorithms to control it.

Algorithm 1. Siphon Control

A monitor $V_S$ is added to $N_i$ such that $S$ is invariant-controlled. The new net is denoted by $(N'_i, M'_{i0})$.

Step 1. Let $\xi_S = 1, M'_{i0}(V_S) = M_{i0}(S) - \xi_S$, where $\xi_S$ is called the control depth variable of $S$.

Step 2. Let $p \in P_{s0} \cap T_{s1}$, add an arc from $V_{s1}$ to $t$ and $W'(V_{s1}, t) = W_{ps}$, where $p \in \{s1 \cup P_{s2} \cup P_{s3} \cup P_{s1} \cup P_{s2} \cup P_{s3} \cup P_{s3} \} = 1$.

Step 3. Let $p \in S$, add an arc from $t$ to $V_{s1}$ and $W'(V_{s1}, t) = W_{p1} - W_{p2}$, where $p \in \{s1 \cup P_{s2} \cup P_{s3} \cup P_{s1} \cup P_{s2} \cup P_{s3} \cup P_{s3} \} = 1$.

Step 4. Let $p \in P_{s0} \cup P_{s2} \cup P_{s3}$, called a source transition. Therefore, in Algorithm 1, the output arcs of the control place are added to the source transitions of $N_i$.

Theorem 9. Let $N_i = \bigcup_{i=1}^{k} N_{i\lambda} = (P_i \cup P_{s0} \cup P_{s2} \cup P_{s3})$, $S = S_{p} \cup S_{R}$ be a STd3PR net and $S$ be a strict minimal siphon. It is invariant-controlled if $S$ is added by Algorithm 1.

Proof. Let $(N'_i, M'_{i0})$ be the new net with $V_S$. Suppose that $S = \{p_1, \ldots, p_k\}$ and $[S] = \{[S](p_{m}) \cdot p_{m} + \ldots + [S](p_{n}) \cdot p_{n}\}$. Let $H(V_S) = S' \cup [S]$. According to Algorithm 1, $S' = \emptyset$ if $[S] \cap P_{s0} = \emptyset$; otherwise $S' = \{p \in S' \cup [S], p \notin S', p \notin S', p \notin S', S \subseteq S' \subseteq S' \cup S' \}$.

Let $S' = \{p_1, \ldots, p_k\}$. Since $V_S \cup H(V_S)$ is the support of a $P$-invariant, $I_1 = [S' \cdot (p_1 \cdot p_1 + \ldots + [S](p_{n}) \cdot p_{n} + \ldots + [S](p_{n}) \cdot p_{n}) + [S](p_{m}) \cdot p_{m} + \ldots + [S](p_{n}) \cdot p_{n} + V_S$ is a $P$-invariant of $N'_i$. Since $S \cup [S]$ is also the support of a $P$-invariant, $I_2 = p_1 + p_2 + \ldots + [S](p_{m}) \cdot p_{m} + \ldots + [S](p_{n}) \cdot p_{n} + V_S$ is a $P$-invariant of $N'_i$. Let $I_1 = I_2 - I_1$. Then $I = p_1 + p_2 + \ldots + [S](p_{m}) \cdot p_{m} + \ldots + [S](p_{n}) \cdot p_{n} - V_S$ is also a $P$-invariant of $N'_i$. That is to say, $I = I_2 - I_1 = I_1^T I_1$ is a $P$-invariant of $N'_i$, where $I_2$ is a $P$-vector of $N_i$ with $\{p|I_2(p) > 0\} \subseteq S$. Note that $I^T M_{i0} = M'_{i0}(S) - \sum_{p \in S'} M_{i0}(p) - M_{i0}(V_S)$. Since $\forall p \in P_{s0} \cup P_{s2} \cup P_{s3}, M_{i0}(p) = M_{i0}(p), \forall p \in P_{s0} \cup P_{s2} \cup P_{s3}, = 0$, we have $I^T M_{i0} = M_{i0}(S) - \xi_S = I_1^T M_{i0} - \xi_S$. By Algorithm 1, we have $\xi_S > 0$.

Due to the fact that the output arcs of the control place are added to the source transitions of $N_i$, no emptiable control-induced siphon is produced.

Take the STd3PR net shown in Fig. 2(b) as an example. Since there is no dependent siphon in the net, one needs to add monitors for strict minimal siphons to prevent deadlocks. According to Algorithm 1, we add $V_{S1}$ for $S_1$, $i = 1, 2, 3$. For $S_1$ with $[S_1] = \{p_{10}^S, p_{3}^S, p_{8}^S\}$, let $M'_{i0}(V_{S1}) = M_{i0}(S_1) - \xi_S = 3 - 1 = 2$. Since $s_1(t_8) \in p_{80}^S$, add the arc $(V_{S1}, t_1)$ and $(V_{S1}, t_8)$ with the weight $W'(V_{S1}, t_1) = W_{p1}^S = 1$ and $W'(V_{S1}, t_8) = W_{p3}^S = 1$. According to Step 3, we add the arcs from $t_1^S$ to $V_{S1}$ with the weight $W'(t_1^S, V_{S1}) = W_{p3}^S - W_{p3}^S = 0$, from $t_4$ to $V_{S1}$ with the weight $W'(t_4, V_{S1}) = W_{p3}^S - W_{p3}^S = 0$, and from $t_9$ to $V_{S1}$ with the weight $W'(t_9, V_{S1}) = W_{p3}^S - W_{p3}^S = 0$. Due to Step 4, the arc from $t_9$ to $V_{S1}$ with the unit weight value is added. Likewise, $V_{S2}, V_{S3}$ can be similarly added.

Theorem 10 ([11]). Let $S$ be a strongly dependent siphon of $N_i$ with $\eta_S = \sum_{i=1}^{n} a_i \eta_{S_i}$ and $S_i$ be the elementary siphons of $S$, $i \in \{1, 2, \ldots, n\}$. $S$ is controlled if $\forall i \in \{1, 2, \ldots, n\}$, $N_s$ is extended by adding a control place $V_{S_i}$ such that $S_i$ is invariant-controlled and (2) $M_{i0}(S) > \sum_{i=1}^{n} a_i (M_{i0}(S_i) - \xi_S)$. 

Theorem 11 ([11]). Let $S$ be a weakly dependent siphon of $N_s$ with $\eta_S = \sum_{i=1}^{n} a_i \eta_{S_i} - \sum_{j=n+1}^{m} a_j \eta_{S_j}$ and $S_j$ be an elementary siphon of $S$, where $i \in \{1, 2, \ldots, n, n+1, \ldots, m\}$. $S$ is controlled if $\forall i \in \{1, 2, \ldots, m\}$, $N_s$ is extended by adding a control place $V_{S_i}$ such that $S_i$ is invariant-controlled and (2) $M_{i0}(S) > \sum_{i=1}^{n} a_i (M_{i0}(S_i) - \xi_S)$. 

© 2010 John Wiley and Sons AsiaPte Ltd and Chinese Automatic Control Society
According to Theorems 10 and 11, if dependent siphon $S$ is not controlled due to Theorems 5, 6, 7 and 8, one can increase the control depth variables of its elementary siphons to make it controlled.

**Algorithm 2.** Deadlock Prevention Policy Based on Elementary Siphons

Let $(N_s, M_{s0})$ with $N_s = \bigcup_{i=1}^{k} N_{si} = (P_s \cup P_s^0 \cup P_{sR} \cup T_s \cup F_s \cup D_s)$ be a STDsP net system.

Input: A marked STDsP net system $(N_s, M_{s0})$

Output: $(N'_s, M'_{s0})$

Step 1. Find the set of elementary siphons $\Pi_E$ and the set of dependent siphons $\Pi_{D}$. Assume that $\Pi_E = \{S_1, S_2, \ldots, S_m\}$ and $\Pi_D = \{DS_1, DS_2, \ldots, DS_l\}$.

Step 2. According to Algorithm 1, add monitors $V_{S_1}, V_{S_2}, \ldots,$ and $V_{S_m}$. The resulting net is denoted by $(N'_s, M'_{s0})$, where $\forall i \in \{1, 2, \ldots, m\}$, $M'_{s0}(V_{S_i}) = M_{s0}(S_i) - \bar{\xi}_{S_i}, \bar{\xi}_{S_i} = 1$.

Step 3. $\Pi_{D} := \emptyset$; $\Pi'_{D} := \emptyset$.

Step 4. $i := 1$.

Step 5. if $i \geq n + 1$, go to Step 7.

Step 6. According to Theorems 5(6), 7(8), and 10(11), if $DS_i$ is controlled then $\Pi'_{D} := \Pi'_{D} \cup \{DS_i\}$, else $i := i + 1$; go to Step 5.

Step 7. $\Pi'_{D} := \Pi_{D} \setminus \Pi'_{D}$.

Step 8. Let $\Pi'_{U} = \{DS_1, DS_2, \ldots, DS_l\}$.

Step 9. $\Pi'_{U} := \emptyset$; $\Pi'_{U} := \emptyset$.

Step 10. $j := 1$.

Step 11. if $j \geq k + 1$, go to Step 13.

Step 12. if $DS_j$ is a well-behaved siphon according to Definition 11, then $\Pi'_{U} := \Pi'_{U} \cup \{DS_j\}$, else $j := j + 1$; go to Step 11.

Step 13. $\Pi'_{U} := \Pi'_{U} \setminus \Pi'_{U}$.

Step 14. if $\forall i \in \{1, 2, \ldots, m\}$, $\forall j \in \{1, 2, \ldots, l\}$, $S_j$ is an elementary siphon of $DS_j$, then $DS_j^{U}(S_j) = 1$; else $DS_j^{U}(S_j) = 0$.

Step 15. Let $\gamma_i = \sum_{DS_j^{U}(S_j) \in \Pi'_{U}} DS_j^{U}(S_j)$, $\forall i \in \{1, 2, \ldots, m\}$.

Step 16. Let $\gamma_{x} = \max \{\gamma_i | i = 1, 2, \ldots, m\}$.

Step 17. Increase $\bar{\xi}_{S_i}$ until every siphon in $\{DS_j^{U}(S_j) \mid DS_j^{U}(S_j) = 1, j \in \{1, 2, \ldots, l\}\}$ is either invariant-controlled due to Theorem 10(11) or well-behaved due to Definition 11.

Step 18. $\Pi'_{U} := \Pi'_{U} \setminus \{DS_j^{U}(S_j) = 1, j \in \{1, 2, \ldots, l\}\}$.

Step 19. if $\Pi'_{U} = \emptyset$, go to Step 21.

Step 20. $\gamma_{x} := 0$, go to Step 16.


In Algorithm 2, $\Pi'_{D}$ is a set of siphons which neither satisfy Theorems 5 and 6 nor are well-behaved.

In order to control the siphons in $\Pi'_{D}$, we have to increase the control depth variables of related elementary siphons such that as many dependent siphons as possible are controlled. Doing so can make all siphons in $\Pi'_{D}$ controlled.

**Lemma 2** ([22]). Let $(N, M_0)$ be a marked net system, $S$ be a strict minimal siphon, and $I$ be a $P$-invariant. $S$ is max-controlled if the following two conditions are satisfied:

1. $\|I\| \subseteq S$, $\|I\| \cap S = \emptyset \text{ or } \forall p \in (\|I\| \cap S), \max_{p^*} = 1$.
2. $I^T \cdot M_0 > \sum_{p \in S} I(p) \cdot (\max_{p^*} - 1)$, where $\max_{p^*} = \max_{p^*} (W(p, t))$.

**Theorem 12.** Let $N_s = \bigcup_{i=1}^{k} N_{si} = (P_s \cup P_s^0 \cup P_{sR} \cup T_s \cup F_s \cup D_s)$ be a STDsP net system and $(N'_s, M'_{s0})$ be the new controlled net according to Algorithm 2. Then every strict minimal siphon in $(N'_s, M'_{s0})$ is max-controlled.

**Proof.** Since no emptiable control-induced siphon is produced in $N'_s$, the siphons in $N'_s$ do not include the additional control places. Therefore the strict minimal siphons in $N'_s$ are also the ones of the original net $N_s$. Even though the controlled net $(N'_s, M'_{s0})$ is a general net, for each strict minimal siphon $S$, $\exists p \in S$ such that $W(p, t) \geq 1$ be true. That is to say, $\forall p \in S$, $\max_{p^*} = 1$.

Due to Algorithm 1, control place $V_S$ can make elementary siphon $S = \{p_1, \ldots, p_l\}$ invariant-controlled by $I = p_1 + \cdots + p_l - [S^T]_1 p_1 - \cdots - [S^T]_l p_l - V_S$. We can get $\|I\| \subseteq S$, $\|I\| \cap S = \emptyset$ and $I^T \cdot M_{s0} - \sum_{p \in S} I(p) \cdot (\max_{p^*} - 1) = M_{s0}(V_S) = \bar{\xi}_{S} \geq 1$.

Due to Lemma 2, every elementary siphon in $N'_s$ satisfies the max-controlled siphon property. According to Algorithm 2, dependent siphons are either well-behaved or invariant-controlled. Note that $\forall p \in S$, $\max_{p^*} = 1$.

The dependent siphons satisfy the max-controlled. As a result, every strict minimal siphon in $N'_s$ is max-controlled.

**Definition 22** ([22]). A net system satisfies the max-controlled siphon property if every strict minimal siphon in it is max-controlled.

**Corollary 1.** The controlled net $(N'_s, M'_{s0})$ satisfies the max cs-property.

**Proof.** The truth of this corollary can be led easily by Theorem 12 and Definition 22.
Lemma 3 ([23]). Let \((N, M_0)\) be a marked \(S^4R\) net. \(N\) is live under \(M_0\) iff \(N\) satisfies the max cs-property.

Theorem 13. Let \((N'_s, M'_s)\) be a marked \(STdS^3PR\) net. It is live under \(M'_s\) iff \(N'_s\) satisfies the max cs-property.

Proof. First the controlled net satisfies the max cs-property. Since each transition in \(N'_s\) has a unit delay time, the factor of delay time can be ignored. In this case, an \(STdS^3PR\) net is actually an \(ES^3PR\) that is a subclass of \(S^4R\) nets. Therefore the result can be reached.

Theorem 14. The controlled \((N'_s, M'_s)\) is live.

Proof. The truth of this theorem can be led easily by Corollary 1 and Theorem 13.

Since the removal of implicit places does not change the behavior and liveness of a net, accordingly Algorithm 3 defined as follows can simplify the structure of a controlled net.

Input: a live controlled \(STdS^3PR\) net system \((N'_s, M'_s)\) with the set of its control places being \(P^s = \{V_1, V_2, \ldots, V_m\} \).
Output: a live controlled \(STdS^3PR\) net system \((N^s, M^s)\) with the set of its control places being \(P^s = \{P^s \mid |P^s| \leq m\} \).

1. \(N := N'_s\).
2. \(M := M'_s\).
3. \(j := 1\).
4. \(P^s := \emptyset\).
5. While \((j \leq m)\) do
6. According to Theorem 3, if \(V_j\) is implicit, then remove \(V_j\) and its related arcs and the resulting net is denoted by \((N'_s, M'_s)\). \(P^s := P^s \cup \{V_j\}\).
7. \(j := j + 1\).
8. End while.
9. \(N^s := N'_s\).
10. \(M^s := M'_s\).
11. Output \((N^s, M^s)\).

By removing the implicit places, the supervisor can be simplified. For example, for the net in Fig. 2(b), three monitors for elementary siphons are added. By Algorithm 3, \(V_5\) is an implicit control place that can be removed. Since the conditions of Theorem 3 is sufficient but not necessary, the number of control places can decrease more and the behavior of the net system may increase after removing the liveness-restricted places.

Input: a live controlled \(STdS^3PR\) net system \((N^s, M^s)\), the set of places \(P^*_s\), and the set of its control places \(P^s = \{V_1, V_2, \ldots, V_k\}\).
Output: a live controlled \(STdS^3PR\) net system \((N^s_A, M^s_A)\) and the set of its control places \(P^s_A\), where \(|P^s_A| \leq k\).

1. \(N := N^s\).
2. \(M := M^s\).
3. \(i := 1\).
4. \(d := |P^s|\).
5. \(\cup := \emptyset\).
6. While \((i \leq k)\) do
7. According to Theorem 4, if \(G^{MIP}(M_i) = d - 1\) then removing \(V_i\) and its related arcs, the result net is denoted by \((N_i, M_i)\). \(N_i := N_s, M_i := M_s, d := d - 1\).
8. \(i := i + 1\).
9. End while.
10. Remove the places in \(\cup\) and their related arcs, the result net is denoted by \((N_i, M_i)\). \(N_A := N_i, M_A := M_i, P^s_A := P^s \setminus \cup\).
11. Output \((N^s_A, M^s_A)\).

Since Theorem 4 is sufficient but not necessary, it is possible that \(V_i\) is still liveness-restricted place if \(G^{MIP}(M_i) < d - 1\). In usual case, however, this algorithm optimizes from the behavior and structure of a controlled net system.

Theorem 15. A live controlled \(STdS^3PR\) net can be reverted into a TdS^3PR. Moreover, the reverted TdS^3PR net is live.

Proof. Apparently, if all the arcs introduced by \(V_5\) are neither input nor output arcs of \(t \in \mathcal{T}\), the SPN can be reverted into a TdPN. According to [4], the two nets have the same liveness properties.

In Algorithm 1, Step 2 shows that for the output node \(t\) of the arcs \((V_5, t)\), we have \(t \in \mathcal{T}\). Step 3 shows that for \(p \in \mathcal{P}\), if \(p^* \in \mathcal{P}_s\), then \(p^* \in \mathcal{S}\) and \(W'(t, V_5) = W_p - W_p' = 0\). In Step 4, in arc \(W'(t, V_5)\), \(t \in p^*\) and \(|p^*| > 1\) are true. We hence have \(t \in \mathcal{T}\) in arc \(W'(t, V_5)\). As a result, in the arcs introduced by \(V_5\) in Algorithm 1, we have \(t \in \mathcal{T}\). That is to say, the \(STdS^3PR\) can be reverted into a TdS^3PR. Since TdPN and its corresponding SPN have the same liveness properties, the reverted TdS^3PR is live.

Algorithm 5. Deadlock Prevention Policy of a TdS^3PR
Input: A marked TdS^3PR that contains deadlocks, denoted by OPN.
Output: A controlled TdS^3PR that is live, denoted by CPN.

1. Stretch the OPN into an SPN.
2. For SPN, use Algorithm 2–4 such that every SMS in SPN is controlled.
3. Revert the controlled SPN to CPN.
Step 4. \( N_f := N_i;\) \( M_f := M_i.\)
Step 5. \( i := i + 1\) if \( i \leq l\) then go to Step 2.
Step 6. Output \((N_f, M_f)\) with \( k\) control places. \( \beta_f\)
denotes the number of reachable markings of \((N_f, M_f)\).
If \( \beta_f = \beta_1\), the controlled behavior of \((N_f, M_f)\) is the
same as \((N_1, M_1)\); otherwise, the controlled behavior of
\((N_f, M_f)\) is more permissive than \((N_1, M_1)\).

Algorithm 7. Back-to-Front removal of redundant control
places for live TdS\(^3\)PR net

Input: a live controlled TdS\(^3\)PR \((N_1, M_1)\) and the set of
its control places \( P_V = \{V_1, V_2, \ldots, V_l\} \). \( \beta_1\) denotes the
number of reachable markings of \((N_1, M_1)\).

Output: a live controlled structurally simple supervisor
of TdS\(^3\)PR with \((N_b, M_b)\) and the set of its control places
\( P_V\), where \(|P_V| \leq 1\).

Step 1. \( N_b := N_i;\) \( M_b := M_1;\) \( i := n;\) \( j := 0;\) \( k := 0.\)
Step 2. Remove \( V_i\) from \((N_b, M_b)\). Denote the resulting
net system by \((N_i, M_i)\). \( \beta_{b1}\) denotes the number of reachable
markings of \((N_i, M_i)\).

Step 3. If \((N_i, M_i)\) is not live, then put \( V_i\) back into \((N_i, M_i)\); \( k := k + 1,\) which means that \( V_i\) is necessary
to keep the net live.

Else if \( \beta_{b1} < \beta_1\), then put \( V_i\) back into \((N_i, M_i)\),
\( k := k + 1;\)
else \( j := j + 1;\)
Endif

Step 4. \( N_b := N_i;\) \( M_b := M_1.\)
Step 5. \( i := i - 1\) if \( i \geq 1\) then go to Step 2.
Step 6. Output \((N_b, M_b)\) with \( k\) control places. \( \beta_b\)
denotes the number of reachable markings of \((N_b, M_b)\).
If \( \beta_b = \beta_1\), the controlled behavior of \((N_b, M_b)\) is the
same as \((N_1, M_1)\); otherwise, the controlled behavior of
\((N_b, M_b)\) is more permissive than \((N_1, M_1)\).

The removal of redundant control places makes
use of both Algorithms 6 and 7. They may reach the
same results or obtain different outcomes.

IV. AN FMS EXAMPLE

The flexible manufacturing cell [15] shown in
Fig. 4 has four machines. Each machine can hold two
parts at the same time. Also the cell contains three
robots and each of them can hold one part. Parts enter
the cell through three loading buffers and leave the cell
through three unloading buffers. The robots deal with
the movements of parts. Three part types are produced.
Their respective production routes can be found in
[11, 15]. The timed Petri net model of the system is
shown in Fig. 4.
Fig. 4. A TdS³PR net model \((N, M_0)\).

The net system is a TdS³PR and contains deadlocks. In Fig. 4, \(t(k)\) denotes the delay time of transition \(t\) is \(k\) units. Fig. 5 is the corresponding stretched net system in which each transition has a unit delay time.

There are 18 strict minimal siphons in Fig. 5 as shown in Table III where dependent SMSs are marked by *. There are only nine elementary siphons among 18 SMSs.

First, Algorithm 1 is applied to the net shown in Fig. 5. A control place is added to each elementary siphon without generating new unmarked siphons. Suppose that \(V_{S_1}\) is the control place for elementary siphon \(S_1 = \{p_{10}, p_{18}, p_{22}, p_{26}, p_1^5, p_5^5, p_1^{10}, p_1^6, p_1^{17}, p_1^{16}, p_2^{16}, p_3^{16}\}\), and the extended net system with \(V_{S_1}\) is denoted by \((N_1', M_{10}')\). Let control depth variable \(\xi_{S_1} = 1\). We have \(M_{10}'(V_{S_1}) = M_{10}(V_{S_1}) - 1 = 2\). It is easy to see that \(I_{p_{26}} = p_{13} + p_{18} + p_{26} + p_1^9 + p_2^9 + p_3^9 + p_1^{10} + p_2^{16} + p_3^{16} + p_1^{17}\), \(I_{p_{22}} = p_{10} + p_{19} + p_{22} + p_1^5 + p_2^5 + p_1^{10} + p_1^{15} + p_2^{15} + p_1^{16} + p_2^{16} + p_3^{16}\). By using linear programming techniques, we first deal with dependent siphon \(S_2\)

\[
\text{min}\{M_1'(S_2)\} \quad \text{subject to} \quad M_1' = M_0' + [N']Y \\
M_1' \geq 0, \quad Y \geq 0
\]

where \(M_0' = [3, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \). It is easy to know that the least number of tokens in \(S_2\) is 2. That is to say, \(\forall M_1' \in R(N_1', M_{10}')\), \(M_1'(S_2) > 0\). Therefore \(S_2\) is a well-behaved siphon. By the same way, we can verify that \(\text{min}\{M_1'(S_3) = 2\}\), \(\text{min}\{M_1'(S_5) = 3\}\), \(\text{min}\{M_1'(S_{11}) = 2\}\), \(\text{min}\{M_1'(S_{12}) = 2\}\), and \(\text{min}\{M_1'(S_{13}) = 2\}\). It implies that \(S_2, S_3, S_5, S_{11}, S_{12}\), and \(S_{13}\) cannot be emptied at any reachable state. Hence there is no need to increase \(\xi\). After adding nine control places, all the siphons are max-controlled. By Algorithm 3, \(V_{S_1}\) and \(V_{S_{14}}\) are implicit. By Algorithm 4, \(V_{S_{15}}\) and \(V_{S_{17}}\) are liveness-restricted places. Removing \(V_{S_7}, V_{S_{14}}, V_{S_{15}}\), and \(V_{S_{17}}\) can reach a structurally simple liveness-enforcing supervisor for the STdS³PR.
Fig. 5. A STdS$^3$PR net ($N_s, M_0$).
Finally, according to Algorithm 5, revert the STdS3PR to a TdS3PR. By Algorithm 6, V_{S18} is a redundant control place. Algorithm 7 obtains the same result as Algorithm 6. The resultant net is denoted by (N_1, M_1) as shown in Fig. 6. It is live and has 5262 reachable states as shown in Table I.

V. CONCLUSION

This research focuses on the deadlock prevention for a class of real-time FMS, where the deadlocks are caused by the unmarked siphons in their Petri net models. The time factor is considered in the net model. The concepts of TdS3PR and STdS3PR nets are introduced in this paper. An FMS is modeled with an TdS3PR which is a special class of timed Petri nets. This paper shows that by stretching the original PN model, time factor can be ignored. Then by adding a monitor for each elementary siphon, testing the controllability of dependent siphons and enlarging the depth variables when necessary, each siphon is successfully max-controlled and no emptiable control-induced siphons can be produced. And then implicit and liveness-restricted places are removed. Finally, by reverting the SPN into a TdPN and removing redundant control places, a liveness-enforcing supervisor with simple structure can be achieved. The advantages of the proposed method can be summarized as follows: (1) less control places and arcs are added to the net to prevent deadlock; (2) it performs offline, which can satisfy the real-time control requirements of a system.

Table I. Control performance comparison.

<table>
<thead>
<tr>
<th>Control Property</th>
<th>Alg. 1</th>
<th>Alg. 2</th>
<th>Alg. 3</th>
<th>Alg. 4</th>
<th>Alg. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Places added</td>
<td>18</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>No. of Arcs added</td>
<td>106</td>
<td>49</td>
<td>37</td>
<td>27</td>
<td>22</td>
</tr>
<tr>
<td>No. of Reachable States</td>
<td>5202</td>
<td>5202</td>
<td>5202</td>
<td>5202</td>
<td>5262</td>
</tr>
</tbody>
</table>

Table II. Marking relationships between dependent and elementary siphons.

<table>
<thead>
<tr>
<th>DS</th>
<th>η relationship</th>
<th>Initial marking relationship</th>
<th>Ctrl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_2^e</td>
<td>η_{S_2} = η_{S_5} + η_{S_5} - η_{S_{10}}</td>
<td>M'(S_2) &lt; M'(S_4) + M'(S_7)</td>
<td>No</td>
</tr>
<tr>
<td>S_3^e</td>
<td>η_{S_1} = η_{S_1} + η_{S_1} - η_{S_{18}}</td>
<td>M'(S_3) &lt; M'(S_4) + M'(S_{14}) - 2</td>
<td>No</td>
</tr>
<tr>
<td>S_4^e</td>
<td>η_{S_1} = η_{S_1} + η_{S_1} - η_{S_{18}}</td>
<td>M'(S_5) &lt; M'(S_7) + M'(S_{14}) - 2</td>
<td>No</td>
</tr>
<tr>
<td>S_6^e</td>
<td>η_{S_0} = η_{S_{10}} + η_{S_{14}}</td>
<td>M'(S_6) &gt; M'(S_{10}) + M'(S_{14}) - 2</td>
<td>Yes</td>
</tr>
<tr>
<td>S_8^e</td>
<td>η_{S_8} = η_{S_{10}} + η_{S_{16}}</td>
<td>M'(S_8) &gt; M'(S_{10}) + M'(S_{16}) - 2</td>
<td>Yes</td>
</tr>
<tr>
<td>S_9^e</td>
<td>η_{S_8} = η_{S_{10}} + η_{S_{16}}</td>
<td>M'(S_5) &gt; M'(S_{10}) + M'(S_{18}) - 2</td>
<td>Yes</td>
</tr>
<tr>
<td>S_{11}^e</td>
<td>η_{S_{11}} = η_{S_{14}} + η_{S_{15}} - η_{S_{16}}</td>
<td>M'(S_{11}) &lt; M'(S_4) + M'(S_{15})</td>
<td>No</td>
</tr>
<tr>
<td>S_{12}^e</td>
<td>η_{S_{12}} = η_{S_{14}} + η_{S_{17}} - η_{S_{18}}</td>
<td>M'(S_{12}) &lt; M'(S_{14}) + M'(S_{17}) - 2</td>
<td>No</td>
</tr>
<tr>
<td>S_{13}^e</td>
<td>η_{S_{13}} = η_{S_{14}} + η_{S_{15}} - η_{S_{16}}</td>
<td>M'(S_{13}) &lt; M'(S_{14}) + M'(S_{15}) - 2</td>
<td>No</td>
</tr>
</tbody>
</table>
Further research includes solving the problems above.

In order to test the controllability of the SMSs in a plant net model, we must find all SMSs. The reachable states are also restricted. Moreover, instantaneous transitions are not considered in this paper. Further research includes solving the problems above and explores optimal policies.

### References


Jinwei Guo received the M.S. degree from Xidian University, Xi’an, China, in 2004. Since 2004, he has been with Xidian University, where he is currently a Lecturer with the School of Electro-Mechanical Engineering. His research interests are Petri net theory and applications, and supervisory control of discrete event systems.

Zhiwu Li received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi’an, China, in 1989, 1992, and 1995, respectively. In 1992, he joined Xidian University, where he is currently a Professor with the School of Electro-Mechanical Engineering. From 2002 to 2003, he was a Visiting Professor with the Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. From February 2007 to February 2008, he was a Visiting Scientist with the
Laboratory for Computer-Aided Design and Lifecycle Engineering, Department of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa. He has over 100 publications, including a book chapter in Deadlock Resolution in Computer-Integrated Systems (Marcel-Dekker, 2005). His main research interests include Petri net theories and applications, supervisory control of discrete event systems, workflow modeling and analysis, and systems integration.


He is a member of the International Advisory Committee of the Tenth International Conference on Automation Technology (Automation 2009), Tainan, Taiwan, June 27–8, 2009. He serves as a frequent reviewer for 15 international journals, including a number of IEEE TRANSACTIONS, as well as many international conferences. Dr. Li is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He is a recipient of Alexander von Humboldt Research Grant from the Alexander von Humboldt Foundation, Germany.