Supervisor Synthesis for Discrete Event Systems with Arbitrary Forbidden State Specifications

Yu Ru, Maria Paola Cabasino, Alessandro Giua and Christoforos N. Hadjicostis

Abstract

In this paper, we consider the forbidden state problem in discrete event systems modeled by Petri nets with uncontrollable and/or unobservable transitions. To handle the interleaving of uncontrollable and unobservable transitions, we first use the reverse net to compute a set of weakly forbidden markings (i.e., a set of markings from which forbidden markings can be reached by firing uncontrollable transitions). We then use basis markings to represent the set of consistent markings for Petri nets with acyclic unobservable subnets (or unobservable subnets with certain cycles). We determine the control policy by checking if a possible subsequent basis marking belongs to the set of weakly forbidden markings; if so, we disable the corresponding (controllable) transition. The setting in this paper generalizes previous work by allowing partial observation, partial control, and a finite number of arbitrary forbidden states.

Published as:

Y. Ru, M.P. Cabasino, A. Giua, C.N. Hadjicostis, "Supervisor Synthesis for Discrete Event Systems with Arbitrary Forbidden State Specifications," CDC08: 47th IEEE Conf. on Decision and Control (Cancun, Mexico), pp. 1048-1053, December 2008.

This work was supported in part by the International Curriculum Option on Hybrid Control for Complex, Distributed and Heterogeneous Embedded Systems (http://www.piaggio.ccii.unipi.it/ICO/) and in part by the National Science Foundation (USA), under NSF Career Award 0092696 and NSF ITR Award 0426831. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF. The research leading to these results has also received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements no. INFSO-ICT-223844 and PIRG02-GA-2007-224877.

Yu Ru is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Christoforos N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus and also with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (e-mails: yuru2@illinois.edu; chadjic@ucy.ac.cy).

Maria Paola Cabasino and Alessandro Giua are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, 09123 Cagliari, Italy (e-mail: cabasino@diee.unica.it; giua@diee.unica.it).

I. Introduction

A discrete event system (DES) is a dynamic system that evolves in accordance with the abrupt occurrence, at possibly unknown and irregular intervals, of physical events [1]. Such systems arise in a variety of contexts, including manufacturing, robotics, vehicular traffic, and computer systems, as well as communication networks.

In many DESs, there may exist system states that are undesirable (e.g., a deadlock state from which there are no further state transitions, or a state that is reached through faulty state transitions). When certain activity in the system can be enabled/disabled, the problem of devising a control strategy to enable/disable transitions so as to avoid forbidden states is called the forbidden state problem and was first introduced by Ramadge and Wonham in [2] in the context of finite automata. Later, this problem was also studied in the Petri net framework. When the set of forbidden states can be represented by linear inequalities (this is possible, for example, when the constraint relates to limited resources in manufacturing systems), many methods apply (e.g., [3]–[6]). Though this assumption is not always viable, there are relatively few works on the control of Petri nets that address arbitrary forbidden state specifications [7]–[10]. Most of these earlier approaches exhibit some limitations: for example, the plant is restricted to cyclic controlled marked graphs in [7] and to bounded Petri nets in [8]–[10].

In this paper we study the arbitrary forbidden state problem in a general Petri net setting, where transitions can be uncontrollable and/or unobservable and the net is not necessarily bounded. The problem is challenging in that (i) there can be interleaving of uncontrollable and unobservable transitions; (ii) possible system states are not unique due to unobservable transitions; (iii) the set of forbidden states can be an arbitrary set of finite cardinality; (iv) it is not clear when and how control actions should be taken since uncontrollable transitions could take place at any time (perhaps without even being observed). Given an arbitrary finite set of forbidden markings, we first compute (offline) the set of weakly forbidden markings which effectively allows us to deal with uncontrollable transitions. An online observer computes basis markings, introduced in [11], [12] to represent consistent markings, and the control policy is determined by checking whether one of the subsequent possible basis markings belongs to the set of weakly forbidden markings. A similar setup was considered in [13] but the authors did not solve the problem of basis marking updating (or reduced state estimate set updating according to the terminology

of [13]). While resolving the above issues, the work in this paper also generalizes the results in [11], [12] in that it allows the use of basis markings to represent the set of consistent markings in unobservable subnets with certain types of cycles; our approach also generalizes the results in [10] by allowing partial observation in Petri nets that are not necessarily bounded.

II. PRELIMINARIES

A. Basic Concepts of Petri Nets

In this subsection we recall notation and basic concepts about Petri nets. For more details, refer to [14].

Definition 1 A Petri net structure is a 4-tuple $N = (P, T, Pre, Post)$, where $P = \{p_1, p_2, ..., p_n\}$ is a set of n places; $T = \{t_1, t_2, ..., t_m\}$ is a set of m transitions; $Pre : P \times T \rightarrow \mathbb{N}_0$ and $Post : P \times T \rightarrow \mathbb{N}_0$ are the pre– and post–incidence functions that specify the arc weights ($\mathbb{N}_0$ is the set of nonnegative integers); $C = Post - Pre$ is the incidence matrix.

The set of all input (or output) places of a transition $t \in T$ is defined as ${}^\bullet t = \{p \in P | Pre(p,t) > 0\}$ (or $t^\bullet = \{p \in P | Post(p,t) > 0\}$). Similarly, the set of all input (or output) transitions of a place $p \in P$ is defined as ${}^\bullet p = \{t \in T | Post(p,t) > 0\}$ (or $p^\bullet = \{t \in T | Pre(p,t) > 0\}$). A Petri net structure is a state machine if for any transition t, $|{}^\bullet t| = |t^\bullet| = 1$ and all arc weights are 1. An elementary cycle of a state machine is a nonempty sequence $x_1 x_2 \cdots x_k$ which satisfies $x_i \in P \cup T$, $x_{i+1} \in x_i^\bullet$ for $i = 1, ..., k-1$ and $x_1 \in x_k^\bullet$, and no x_i occurs more than once in the sequence [15]. A Petri net structure is acyclic if it has no directed circuits.

A marking is a vector $M : P \rightarrow \mathbb{N}_0$ that assigns to each place a nonnegative integer number of tokens. Pictorially, places are represented by circles, transitions by bars and tokens by black dots. We use $M(p)$ to denote the number of tokens in place p. A Petri net $G = \langle N, M_0 \rangle$ is a net structure N with an initial marking M_0.

A transition t is state-enabled at marking M if $M \geq Pre(\cdot, t)$, which is denoted by $M[t\rangle$. A state-enabled transition t may fire yielding the marking $M' = M + C(\cdot, t)$, where $C(\cdot, t)$ denotes the column of matrix C that corresponds to transition t. In this paper, we assume that only one transition can fire at any instant. A k-length firing sequence from marking M is a sequence of transitions $\sigma = t_{s_1} t_{s_2} \cdots t_{s_k}$, $t_{s_i} \in T$, such that $M[t_{s_1}\rangle M_1[t_{s_2}\rangle M_2 \cdots [t_{s_k}\rangle M'$; this is

denoted by $M[\sigma\rangle M'$ and we say σ is state-enabled at marking M. The final marking M' can also be calculated by the following state equation

$$M' = M + C\vec{y}, \tag{1}$$

where $\vec{y} \in \mathbb{N}_0^m$ is called the firing vector of σ and satisfies $\vec{y}(i) = k_i$ if transition t_i appears k_i times in σ. The mapping from σ to $\vec{y}$ is denoted by $\pi : T^* \to \mathbb{N}_0^m$, i.e., $\vec{y} = \pi(\sigma)$. An m-dimensional vector $\vec{x}$ of nonnegative integers is called a transition invariant if $C\vec{x} = \vec{0}$, indicating that the firing of a sequence of transitions with firing vector $\vec{x}$ leaves the marking of the Petri net unchanged.

A marking M is reachable in $\langle N, M_0\rangle$ if there exists a firing sequence σ such that $M_0[\sigma\rangle M$. The set of all markings reachable from M_0 defines the reachability set of $\langle N, M_0\rangle$ and is denoted by $R(N, M_0)$. A Petri net $\langle N, M_0\rangle$ is bounded if there exists a positive constant K such that $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq K$. A Petri net is structurally bounded if it is bounded for any initial marking.

B. Petri Nets with Uncontrollable and/or Unobservable Transitions

We assume that the set of transitions T is partitioned in two distinct ways: i) $T = T_c \cup T_{uc}$ and $T_c \cap T_{uc} = \emptyset$, in which T_c (or T_{uc}) consists of all controllable (or uncontrollable) transitions; ii) $T = T_o \cup T_{uo}$ and $T_o \cap T_{uo} = \emptyset$, in which T_o (or T_{uo}) consists of all observable (or unobservable) transitions. Uncontrollable transitions are transitions that cannot be disabled by a supervisor. For example, state transitions in chemical reactions are usually uncontrollable, and actuator (or other) failures can also be modeled by uncontrollable transitions. Unobservable transitions are transitions that cannot be directly observed given current sensor availability (no sensors exist for such transitions). In this paper, we adopt the common assumption that $T_c \subseteq T_o$ (or equivalently, $T_{uo} \subseteq T_{uc}$) [13].

The firing of each observable transition t causes a sensor to generate a unique label t; however, the firing of an unobservable transition goes unrecorded. More formally, we define the observation mask $P_o : T^* \to T_o^*$ as i) $P_o(\varepsilon) = \varepsilon$, where ε is the empty string; ii) for all $\sigma \in T^*$ and $t \in T$, $P_o(\sigma t) = P_o(\sigma)t$ if $t \in T_o$ and $P_o(\sigma t) = P_o(\sigma)$ otherwise. Finally, we use m_o (or m_{uo}) to denote the cardinality of set T_o (or T_{uo}), and C_o (or C_{uo}) to denote the restriction of the incidence matrix to T_o (or T_{uo}).

Definition 2 Given a Petri net $G = \langle N, M_0 \rangle$ with $N = (P, T, Pre, Post)$ and $T = T_o \cup T_{uo}$, we define the set of markings that are consistent with a sequence of observed labels $\omega \in T_o^*$ as $\mathcal{C}(\omega) = \{M \in \mathbb{N}_0^n | \exists \sigma \in T^* : M_0[\sigma\rangle M, P_o(\sigma) = \omega\}$.

Remark 1 Note that since we assume $T_c \subseteq T_o$, the set $\mathcal{C}(\omega)$ only depends on ω and not on the history of control actions that may have been taken (in general, if some unobservable transitions are controllable, then the history of control actions can influence $\mathcal{C}(\omega)$ via the enabling/disabling of unobservable but controllable transitions).

To handle uncontrollable and unobservable transitions, we need the concepts of the T'-induced subnet [16] and the reverse net [17].

Definition 3 Given a net structure $N = (P, T, Pre, Post)$, and a set of transitions $T' \subseteq T$, we define the T'-induced subnet of N as $N_{T'} = (P, T', Pre', Post')$ where Pre' (or $Post'$) is the restriction of Pre (or $Post$) to $P \times T'$.

In this paper, we use the T_{uc}-induced subnet (also called uncontrollable subnet) and the T_{uo}-induced subnet (also called unobservable subnet).

Definition 4 Given a net structure $N = (P, T, Pre, Post)$, $N' = (P, T, Pre', Post')$ is said to be its reverse net if $Pre' = Post$ and $Post' = Pre$.

III. Problem Formulation

In the forbidden state problem we consider, the system is modeled by a Petri net $G = \langle N, M_0 \rangle$ with uncontrollable transitions T_{uc} and/or unobservable transitions T_{uo}. Our goal is to determine a maximally permissive control policy (which is defined shortly) based on the observation of a sequence of observable transitions ω such that the system is guaranteed to avoid entrance to any state in a finite set of forbidden states M_F. Moreover, we assume that the net G satisfies the following assumptions:

A1 the unobservable subnet (namely, the T_{uo}-induced subnet) is acyclic;

A2 the reverse net of the uncontrollable subnet (namely, the reverse net of the T_{uc}-induced subnet) is structurally bounded and the set of forbidden markings M_F has finite cardinality;

A3 $T_{uo} \subseteq T_{uc}$.

To define the supervisory control policy, we take the set of possible control actions at the given point (i.e., after having observed sequence ω) to be all possible subsets of controllable transitions. More formally, we define the control set as $U = \{u \mid u \subseteq T_c\}$, where u is called a control value. A controllable transition t is said to be control-enabled[1] if $t \in u$. If a transition t is both state-enabled and control-enabled, it is enabled and can fire following the state equation (1).

A supervisory control policy f is a function $f : R(G) \rightarrow U$ which specifies the control value u at any reachable marking $M \in R(G)$. To handle unobservable transitions, we need to extend the definition of f from a single marking to a set of consistent markings. As several control policies may realize the same control goal, we try to find the maximally permissive control policy. In this paper, we adopt the optimality criterion in [13], [18], i.e., for each set of markings $\mathcal{C}(\omega)$ (consistent with the observation of sequence ω), we want the control value $f(\mathcal{C}(\omega))$ to be such that the set of enabled transitions is as large as possible while ensuring that forbidden markings will never be visited. Therefore, given an observed sequence of transitions $\omega \in T_o^*$, the control policy is defined as $f(\mathcal{C}(\omega)) \subseteq U$. Note that control action at the current step applies to any marking consistent with ω because all unobservable transitions are allowed to fire after the observation of ω (as $T_{uo} \subseteq T_{uc}$). Therefore, our definition of control policy $f(\mathcal{C}(\omega))$ is consistent regardless of the previous control decisions that have been made.

IV. Characterization of Consistent Markings

To determine the control policy f, we need to know all possible current states (namely, $\mathcal{C}(\omega)$) given an observed sequence of labels ω. One simple idea is to enumerate all these markings (the number of consistent markings increases at most polynomially in the length of the observed sequence of labels [19]). However, under Assumption **A1** given in Section III, the set of consistent markings can be characterized more concisely using a subset of consistent markings called basis markings [11], [12]. A relaxation of Assumption **A1** is discussed in Section VI.

Definition 5 Let $G = \langle N, M_0 \rangle$ be a Petri net with unobservable transitions T_{uo}. Given a sequence ω of observed transition labels, a *basis marking* $M_{b,\omega}$ is a marking reached from the initial marking M_0 by firing ω and all those unobservable transitions that are strictly necessary to enable ω.

[1]In this paper, uncontrollable transitions are always control-enabled by convention.

Under Assumption **A1**, given any observed sequence of labels ω, it is possible to determine a set of basis markings $\mathcal{M}_{b,\omega}$ such that the set of consistent markings satisfies

$$C(\omega) = \{M \in \mathbb{N}_0^n \mid \exists M_{b,\omega} \in \mathcal{M}_{b,\omega}, \\ \exists \sigma \in T_{uo}^* : M_{b,\omega}[\sigma\rangle M\} \tag{2}$$

(c.f. Theorem 4.7 of [12]). The result is important in that basis markings can be used not only to represent the set of consistent markings but also to determine the control policy. The latter will become clearer in Section V.

Now we consider how to compute the set of basis markings given an observed sequence of labels. First, we need the concept of minimal explanations.

Definition 6 Given a marking M and an observable transition $t \in T_o$, we define the set of explanations of t at M as

$$\Sigma(M,t) = \{\sigma \in T_{uo}^* \mid M[\sigma\rangle M',\ M' \geq Pre(\cdot,t)\}$$

and we define

$$Y(M,t) = \{\vec{e} \in \mathbb{N}_0^{m_{uo}} \mid \exists \sigma \in \Sigma(M,t)\ :\ \pi(\sigma) = \vec{e}\}$$

as the e-vectors (or explanation vectors), i.e., the firing vectors associated to the explanations. Similarly, we define the set of minimal explanations of t at M as

$$\Sigma_{\min}(M,t) = \{\sigma \in \Sigma(M,t) | \nexists\ \sigma' \in \Sigma(M,t) : \pi(\sigma') \lneqq \pi(\sigma)\}$$

and we define

$$Y_{\min}(M,t) = \{\vec{e} \in \mathbb{N}_0^{m_{uo}} | \exists\ \sigma \in \Sigma_{\min}(M,t) : \pi(\sigma) = \vec{e}\}$$

as the corresponding set of minimal e-vectors.

The following algorithm [11], [12], when applied to nets whose unobservable subnet is acyclic, computes the set $Y_{\min}(M,t)$ and terminates after finding all vectors in $Y_{\min}(M,t)$.

Algorithm 1 [Computation of $Y_{\min}(M,t)$ in acyclic unobservable subnets]

1. Let $\Gamma := \left|\begin{array}{c|c} C_{uo}^T & I_{m_{uo}\times m_{uo}} \\ \hline A & B \end{array}\right|$

where $A := (M - Pre(\cdot,t))^T$, $B := \vec{0}_{m_{uo}}^T$.

2. While $A \geq \mathbf{0}^T$

2.1 Choose an element $A(i^*, j^*) < 0$.

2.2 Let $\mathcal{I}^+ = \{i \mid C_{uo}^T(i, j^*) > 0\}$.

2.3 For all $i \in \mathcal{I}^+$

2.3.1 add to $[A \mid B]$ a new row

$[A(i^*, \cdot) + C_{uo}^T(i, \cdot) \mid B(i^*, \cdot) + \vec{e}_i{}^T]$

where $\vec{e}_i$ is the i-th canonical basis vector.

2.4 Remove the row $[A(i^*, \cdot) \mid B(i^*, \cdot)]$ from the table.

End while

3. Remove from B any row that covers other rows.

4. Each row of B is a vector in $Y_{\min}(M, t)$.

Using Algorithm 1, the set of basis markings can be computed recursively as follows: initialize $\mathcal{M}_{b,\varepsilon} = \{M_0\}$, and $\forall \omega \in T_o^*$, $\forall t \in T_o$, calculate $\mathcal{M}_{b,\omega t}$ via the recursion

$$\mathcal{M}_{b,\omega t} = \{M \mid \exists M' \in \mathcal{M}_{b,\omega}, \exists \vec{e} \in Y_{\min}(M', t) : M = M' + C(\cdot, t) + C_{uo}\vec{e}\}. \tag{3}$$

V. Supervisor Synthesis

A. Existence of Maximally Permissive Supervisor

The presence of uncontrollable transitions complicates the forbidden state problem because we also need to prevent the current state from reaching certain states that are not explicitly forbidden. As we will see, once we enter such states, we will never be able to control the system and ensure that the current state is legal. We call such markings weakly forbidden markings [7].

Definition 7 Given a Petri net G with uncontrollable transitions T_{uc} and unobservable transitions T_{uo} such that $T_{uo} \subseteq T_{uc}$, and given a set of forbidden markings M_F, the set of *weakly forbidden markings* with respect to M_F is given by $W(M_F) = \{M \mid \exists M' \in M_F, \exists \sigma \in T_{uc}^* : M[\sigma\rangle M'\}$.

The set of weakly forbidden markings $W(M_F)$ can be computed using the following proposition (c.f. [8]).

Proposition 1 Given a Petri net G with uncontrollable transitions T_{uc} and unobservable transitions T_{uo} (such that $T_{uo} \subseteq T_{uc}$), and given a set of forbidden markings M_F, the set of weakly forbidden markings is given by

$$W(M_F) = \bigcup_{M \in M_F} R(N'_{T_{uc}}, M),$$

where $N'_{T_{uc}}$ is the reverse net of the T_{uc}-induced subnet.

Following Proposition 1, we can obtain $W(M_F)$ by computing all markings reachable from M_F in the reverse net of the T_{uc}-induced subnet. The computation of $W(M_F)$ may be complicated as $W(M_F)$ is not necessarily finite and the computation essentially involves reachability analysis. However, under Assumption **A2** made in Section III, the set of weakly forbidden markings is finite.

The existence of the maximally permissive control policy can be determined by checking whether $M_0 \notin W(M_F)$, which can be directly deduced from Theorem 2 in [20]. More specifically, if $M_0 \notin W(M_F)$, then there is a maximally permissive control policy. Note that this condition is also necessary. That is to say, if $M_0 \in W(M_F)$, there is no control policy which can *guarantee* that the system will never reach a forbidden state.

Now we consider how to check if $M_0 \in W(M_F)$. Suppose the number of weakly forbidden markings is l. One method is to compare M_0 with every marking M in $W(M_F)$; it will take $n \times l$ comparisons, which can be problematic if l is large. Another way to check whether $M_0 \in W(M_F)$ is to use a binary search algorithm [21]. Before searching, we can sort the markings in $W(M_F)$ elementwise: first sort these markings in ascending order of their first components; then sort markings that have the same first components in ascending order of their second components; and so on. Having sorted the markings in $W(M_F)$, we can examine if the initial marking M_0 is in the set of weakly forbidden markings using a modified binary search algorithm (we call it *componentwise binary search*). To analyze the complexity of this approach, let x_1 denote the number of markings that have the same first component $M_0(p_1)$, x_2 denote the number of markings that have the same first component $M_0(p_1)$ and the same second component $M_0(p_2)$, ..., and x_n denote the number of markings that are identical to M_0. Note that[2] x_n is either 1 (i.e., $M_0 \in W(M_F)$) or 0 (i.e., $M_0 \notin W(M_F)$). We search the first component of M_0 in the sorted

[2]This is the worst case scenario as it might be the case that $x_i = 0$ for some $i < n$.

marking set according to their first components using the binary search algorithm and this can be done in at most[3] $\log_2(l-(x_1-1))+1$ comparisons; we then search the second component of M_0 using the binary search algorithm only in the sorted markings that have the same first components as $M_0(p_1)$ and this can be done in at most $\log_2(x_1-(x_2-1))+1$ comparisons; we keep doing this until the last component. The total number of comparisons needed is upper bounded by $B=\log_2(l-(x_1-1))+\log_2(x_1-(x_2-1))+\cdots+\log_2(x_{n-1}-(x_n-1))+n$. Using the arithmetic-mean/geometric-mean inequality[4] [22], we have

$$\begin{aligned} B &= \log_2((l-(x_1-1))\cdot(x_1-(x_2-1))\cdots \\ &\quad \cdot(x_{n-1}-(x_n-1)))+n \\ &\leq n\log_2(\frac{l+n}{n})+n \end{aligned}$$

which is $\mathcal{O}(n\log l)$ if $l\gg 1$.

Note that the resulting $\mathcal{O}(n\log l)$ complexity does not depend on the specific values of $x_1,\cdots,x_n$. Also note that we did not take into account the effort spent in sorting the set of weakly forbidden markings because this componentwise binary search strategy will also be used online when we determine the control policy (and therefore, the cost of sorting is amortized over all these computations as well).

B. Determination of Control Policy

In this subsection, we determine the maximally permissive supervisor using the state estimate (namely, basis markings) and componentwise binary search.

The basic idea is the following. Given the sequence of (observable) transitions seen so far, we keep track of the set of basis markings. At the current step, we compute the set of the subsequent basis markings for each controllable transition t and we disable t if at least one of the subsequent basis markings is in the set of weakly forbidden markings (recall that the control action at the current step determines which controllable transition to enable – this way we are guaranteed that a controllable transition that we meant to disable does not take place). Once a

[3]The last term 1 takes special cases into account (e.g., $x_1=l$, or $x_1=l-1$, etc).

[4]For any n nonnegative real numbers $x_1,x_2,\ldots,x_n$, the arithmetic-mean/geometric-mean inequality is $\frac{\sum_{i=1}^n x_i}{n}\geq\sqrt[n]{\prod_{i=1}^n x_i}$.

new transition label is observed, we update the set of basis markings based on this label and then determine the next control action. The complete two-stage algorithm is given below.

Algorithm 2 [Supervisor Synthesis]

Input: A Petri net G with uncontrollable transitions T_{uc}, unobservable transitions T_{uo} (such that $T_{uo} \subseteq T_{uc}$), a finite set of forbidden markings M_F, and a streaming sequence of observed labels ω.

Output: A control value u at each observation.

Stage 1: Offline Checking of Supervisor Existence

1) Compute the set of weakly forbidden markings $W(M_F)$ using the reverse net of the T_{uc}-induced subnet.

2) Check if $M_0 \in W(M_F)$ using componentwise binary search. If $M_0 \notin W(M_F)$, the supervisor exists; else, exit.

Stage 2: Online Determination of Control Policy

1) Let $\omega = \varepsilon$ and $\mathcal{M}_{b,\omega} = \{M_0\}$.

2) Let $T_F = \emptyset$.

For every $t \in T_c$

Compute $\mathcal{M}_{b,\omega t}$ based on $\mathcal{M}_{b,\omega}$ using Eq. (3).

$\forall M$ appearing in $\mathcal{M}_{b,\omega t}$, if $M \in W(M_F)$ (this can be checked using componentwise binary search), then $T_F = T_F \cup \{t\}$.

3) Output the control value $u = T_c \backslash T_F$ at the current step.

4) Wait until an observable transition t fires.

5) Compute $\mathcal{M}_{b,\omega t}$ based on $\mathcal{M}_{b,\omega}$ using Eq. (3); let $\omega = \omega t$.

6) Goto Step 2.

At Step 2 in Stage 2, we determine the control value at the current step by examining if some subsequent basis marking is a weakly forbidden marking; at this step, we can also store $\mathcal{M}(\omega t)$ if $t \in u$ so that we do not need to recompute it at Step 5 when/if the subsequent observed label is t.

We show the correctness of the algorithm by proving the following two facts:

1) $\forall t \in u$ given at Step 3 in Stage 2, the firing of t will not drive the system from a legal state to any forbidden state. Suppose there exists a marking M consistent with ωt (namely $M \in \mathcal{C}(\omega t)$) such that $M \in W(M_F)$; then, there exists a basis marking $M_{b,\omega t}$, from which M is reached by firing a sequence of unobservable transitions, such that $M_{b,\omega t}$ is in $W(M_F)$ based on the definition of $W(M_F)$. Therefore, t should have been disabled according to Step 2 in Stage 2; a contradiction. The fact shows that the control policy is permissive.
2) $\forall t \in T_c \backslash u$, the firing of t can in fact result in a forbidden state because there exists a marking M, which is consistent with ωt (actually it is a basis marking) and is in $W(M_F)$, i.e., it can uncontrollably reach a forbidden marking. This fact shows that the control policy is maximally permissive.

VI. Extension to Other Unobservable Subnets

In Section III, the unobservable subnet was assumed acyclic. Under this assumption, we can use Algorithm 1 to calculate minimal e-vectors so that we can compute a set of basis markings. However, intuitively, we can also compute the set of such basis markings even for unobservable subnets with cycles, as illustrated in the following example.

Example 1 We consider the unobservable subnet shown in Fig. 1. Suppose the initial state of the subnet is $M_0 = (1\ 0\ 0\ 0\ 2\ 0)^T$ and t is an observable transition. The set of minimal explanations $\Sigma_{\min}(M_0, t)$ as defined in Section IV cannot be found using Algorithm 1 directly because this subnet is not acyclic. It is easy to see that the only minimal explanation is $t_4 t_2$, which results in the basis marking $M = (0\ 0\ 0\ 0\ 1\ 0)^T$. It can also be verified that the set of markings consistent with $\sigma = t$ can be represented using Eq. (2). If we examine the unobservable subnet in Fig. 1 more carefully, there is a transition invariant $(0\ 0\ 1\ 1\ 1)^T$. Thus, any minimal e-vector should *not* cover the invariant based on the definition of minimal e-vectors. ■

One way to generalize the idea in Example 1 is to use the following assumption in place of **A1**:

A1′ the unobservable subnet is a state machine.

Note that an elementary cycle in a state machine represents a transition invariant $\vec{x}$ because for any transition in a state machine, there is exactly one input place and one output place (with all associated arcs having unit weight). Therefore, any minimal e-vector $\vec{e}$ should not cover such

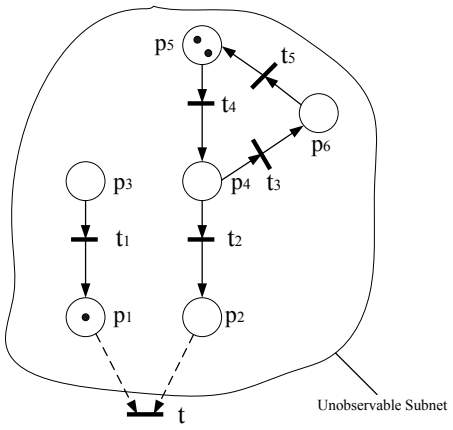

Fig. 1. Unobservable subnet with cycles.

a transition invariant $\vec{x}$ (otherwise, the e-vector is not minimal). This condition can be used as a criterion to stop further calculation when we try to apply Algorithm 1 to Petri nets of which unobservable subnets are state machines.

Before we modify Algorithm 1, we first compute all elementary cycles in the unobservable subnet under Assumption **A1′**. Assume there are q such cycles and denote the set of firing (binary) vectors corresponding to these cycles by $\Gamma = \{\vec{\gamma}_1, \vec{\gamma}_2, \cdots, \vec{\gamma}_q\}$, i.e., $\vec{\gamma}_j(i) = 1$ if transition t_i appears in cycle j and $\vec{\gamma}_j(i) = 0$ otherwise. Note that $\vec{\gamma}_j$ is a transition invariant for $j = 1, 2, \cdots, q$. The modified algorithm is given below.

Algorithm 3 [Computation of $Y_{\min}(M,t)$ in unobservable subnets that are state machines]

1. Let $\Gamma := \left|\begin{array}{c|c} C_{uo}^T & I_{m_{uo}\times m_{uo}} \\ \hline A & B \end{array}\right|$

where $A := (M - Pre(\cdot,t))^T$, $B := \vec{0}_{m_{uo}}^T$.

2. While $A \geq \mathbf{0}^T$

2.1 Choose an element $A(i^*, j^*) < 0$.

2.2 Let $\mathcal{I}^+ = \{i \mid C_{uo}^T(i, j^*) > 0\}$.

2.3 For all $i \in \mathcal{I}^+$

2.3.1 compute a row

$[A(i^*,\cdot) + C_{uo}^T(i,\cdot) \mid B(i^*,\cdot) + \vec{e}_i^{\,T}]$. If there does not exist a $j \in \{1, \ldots, q\}$ such that

$B(i^*, \cdot) + \vec{e}_i^T \geq \vec{\gamma}_j$, then add the row to $[A \mid B]$.

2.4 Remove the row $[A(i^*, \cdot) \mid B(i^*, \cdot)]$ from the table.

End while

3. Remove from B any row that covers other rows.

4. Each row of B is a vector in $Y_{\min}(M, t)$.

In Algorithm 3, we simply check if the temporary e-vector covers some $\vec{\gamma}_j$; if it indeed covers one, we can discard it because it cannot produce a minimal explanation. The algorithm stops in a finite number of steps because the firing vector of the rows we add cannot contain repetitive components in Γ and hence are in finite number.[5] The fact that Algorithm 3 gives all minimal e-vectors can be proved in a manner similar to the way Algorithm 1 was proved in [11].

Theorem 1 Given a Petri net satisfying Assumption **A1′** and an observed sequence of labels ω, a set of basis markings $\mathcal{M}_{b,w}$ can be calculated using Algorithm 3 and Eq. (3) such that the set of consistent markings is

$$\mathcal{C}(\omega) = \{M \in \mathbb{N}_0^n \mid \exists M_{b,\omega} \in \mathcal{M}_{b,\omega}, \exists \sigma \in T_{uo}^* : M_{b,\omega}[\sigma\rangle M\}.$$

We just give a sketch of the proof. i) If the T_{uo}-induced subnet is an acyclic state machine, then the result is a direct application of Theorem 12 in [12]. ii) If the T_{uo}-induced subnet is a state machine with cycles, we observe (see Theorem 6.2 in [23]) that in a state machine a marking M is reachable from M_0 with a sequence σ iff it is also reachable with a sequence σ' where $\pi(\sigma') \leq \pi(\sigma)$ and $\pi(\sigma') \not\geq \vec{x}$ for all transition invariants $\vec{x}$. In other words, if a marking is reachable it is also reachable with a minimal firing sequence whose firing subnet is acyclic.[6] Thus the result of Theorem 12 in [12] also applies to T_{uo}-induced subnets that are state machines.

Theorem 1 shows that the set of consistent markings for Petri nets with unobservable subnets that are state machines can also be represented using basis markings. Therefore, the results in

[5] Another way to realize this is the following: as a minimal e-vector cannot cover the vector corresponding to any elementary cycle, the algorithm is equivalent to computing minimal e-vectors in some acyclic state machine. Therefore, the algorithm must terminate.

[6] The firing subnet of a net is the subnet containing all transitions $\{t \in T \mid \sigma(t) \geq 0\}$ and all their input output places [23].

Section V can also be applied to Petri nets satisfying Assumptions **A1**$'$, **A2** and **A3**.

VII. Conclusions

In this paper, we consider the arbitrary forbidden state problem in DESs modeled by Petri nets. Under the assumption that $T_{uo} \subseteq T_{uc}$ and some conditions on the unobservable subnet and the uncontrollable subnet, we show that basis markings can be used not only to represent the set of consistent markings but also to determine the control policy. We also propose an online algorithm that utilizes componentwise binary search to determine the control policy based on the sequence of observed labels.

Note that in this paper, there is no structure on the set of forbidden markings; if there is, we might be able to compute the set of weakly forbidden markings more efficiently. Also note that there might be deadlocks in the system under the control policy. Investigating these aspects will be part of our future work.

References

[1] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, pp. 81–98, Jan. 1989.

[2] ——, "Modular feedback logic for discrete event systems," *SIAM J. Control and Optimization*, vol. 25, pp. 1202–1218, Sep. 1987.

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

[4] H. Chen, "Control synthesis of Petri nets based on S-decreases," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 10, pp. 233–249, 2000.

[5] G. Stremersch and R. K. Boel, "Structuring acyclic Petri nets for reachability analysis and control," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 12, pp. 7–41, 2002.

[6] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 137–142, Feb. 2003.

[7] L. E. Holloway and B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 35, pp. 514–523, May 1990.

[8] Y. Ru, W. Wu, H. Su, and J. Chu, "Supervisor synthesis for bounded Petri nets based on a transformation function," in *Proc. of 2004 American Control Conference*, Boston, USA, June 2004, pp. 4493–4498.

[9] Z. Achour, N. Rezg, and X. Xie, "On the existence of Petri net controller for discrete event systems under partial observation," in *Proc. of 16th IFAC World Congress 2005*, Jul. 2005.

[10] Y. Ru and C. N. Hadjicostis, "Fault-tolerant supervisory control of discrete event systems modeled by bounded Petri nets," in *Proc. of 2007 American Control Conference*, New York, USA, Jul. 2007, pp. 4945–4950.

[11] A. Giua and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," in *44th IEEE Conf. on Decision and Control*, Seville, Spain, Dec. 2005, pp. 6323–6328.

[12] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica (preliminarily accepted)*, 2008.

[13] L. Zhang and L. E. Holloway, "Forbidden state avoidance in controlled Petri nets under partial observation," in *Proc. 33rd Allerton Conf.*, Monticello, Illinois, Oct. 1995, pp. 146–155.

[14] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541–580, Apr. 1989.

[15] J. Desel and J. Esparza, *Free Choice Petri Nets*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[16] D. Corona, A. Giua, and C. Seatzu, "Marking estimation of Petri nets with silent transitions," in *43rd IEEE Conf. on Decision and Control*, Atlantis, Bahamas, Dec. 2004, pp. 966–971.

[17] H. J. Genrich and E. Stankiewicz-Wiechno, "A dictionary of some basic notions of net theory," *Lecture Notes in Computer Science, Vol. 84: Net Theory and Applications*, pp. 519–535, 1980.

[18] G. Stremersch, *Supervision of Petri Nets*. Kluwer Academic Publishers, 2001.

[19] Y. Ru and C. N. Hadjicostis, "Bounds on the number of markings consistent with label observations in Petri nets," *to appear in IEEE Trans. on Automation Science and Engineering*, 2008.

[20] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net methods for controlled discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 7, pp. 151–190, 1997.

[21] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*. New York, USA: Prentice Hall Software Series, 1988.

[22] E. F. Beckenbach and R. E. Bellman, *Inequalities*. New York, USA: Springer-Verlag, 1965.

[23] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, USA, 1992.