On Supervisory Policies that Enforce Liveness in Completely Controlled Petri Nets with Directed Cut-Places and Cut-Transitions

Ramavarapu S. Sreenivas

***Abstract*—The process of synthesizing a supervisory policy that enforces liveness in a *Petri net* (PN), where each transition can be prevented from firing by an external agent, can be computationally burdensome in general. In this paper we consider PN's that have a *directed cut-place* or a *cut-transition*. A place (transition) in a connected PN is said to be a *cut-place* (*cut-transition*) if its removal will result in two disconnected component PN's. A cut-place is said to be a *directed cut-place*, if in the original PN, all arcs into this cut-place emanate from transitions in only one of the two disconnected component PN's. The authors show there is a supervisory policy that enforces liveness in the original PN if and only if similar policies exist for two PN's derived from the disconnected components obtained after the removal of the directed cut-place (cut-transition). The utility of this observation in alleviating the computational burden of policy synthesis is illustrated via example.**

***Index Terms*—DEDS, liveness, Petri nets, supervisory control.**

I. Introduction

Petri nets (PN's) [3], [4] are a popular modeling paradigm for a wide class of discrete-event dynamic systems (DEDS's). Typically, transitions in a PN represent events, while the token loads of the places indicate the various logical or symbolic conditions in the DEDS. We seek DEDS's with the property that from any reachable state, every event must be executable, although not necessarily immediately. This property is referred to as *liveness*.

In a *live* PN, from every reachable marking it should be possible to fire any transition, although not necessarily immediately. References [7]–[9] concern the existence and synthesis of supervisory policies that enforce liveness in PN's that are not live. In this paper it is assumed that each transition in the PN can be prevented from firing by an external agent, the supervisor. In the general case, the synthesis of supervisory policies that enforce liveness in an arbitrary PN can be computationally burdensome [7]. In this paper we present procedures of alleviating this computational burden when the plant PN has a *directed cut-place* or a *cut-transition*. A place (transition) in a connected PN is said to be a *cut-place* (*cut-transition*) if its removal results in two disconnected component PN's. A cut-place is said to be a *directed cut-place* if all directed arcs into this cut-place emanate from transitions that belong to only one of the above-mentioned component PN's. It is shown that there is a supervisory policy that enforces liveness in the original plant PN, if and only if similar policies exist for two PN's derived from the component PN's mentioned earlier. The utility of this observation in reducing the computational burden of synthesis of supervisory policies is illustrated via an example. Directed cut-places are frequently encountered in PN models of the flow of objects/entities in producer-consumer systems (cf. [1, Sec. I-F]) and systems that use binary semaphores (cf. [1, Ch. 4]). PN models of product-systems shared-events (cf. [5, Sec. X]) typically contain cut-transitions. The next section presents the appropriate notational preliminaries.

Manuscript received September 30, 1997. Recommended by Associate Editor, F. Lin. This work was supported in part by the National Science Foundation under Grant ECS-9409691.

The author is with the Coordinated Science Laboratory and the Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: rsree@uiuc.edu).

Publisher Item Identifier S 0018-9286(99)03949-5.

II. Notational Preliminaries and Review of Prior Work

A PN $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$ is an ordered four-tuple, where $\Pi = \{p_1, p_2, \cdots, p_n\}$ is a set of n *places*, $T = \{t_1, t_2, \cdots, t_m\}$ is a set of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, $\boldsymbol{m}^0 : \Pi \rightarrow \mathcal{N}$ is the *initial-marking function* (or the *initial-marking*), and $\mathcal{N}$ is the set of nonnegative integers. The *marking* of a PN, $\boldsymbol{m} : \Pi \rightarrow \mathcal{N}$, identifies the number of *tokens* in each place. A PN N is said to be *connected* if for any two members $x, y \in \Pi \cup T$, there is either a path from x to y or from y to x. Without loss in generality, in this paper we consider connected PN's only. For a given marking $\boldsymbol{m}$ a transition $t \in T$ is said to be *enabled* if $\forall p \in ({}^\bullet t)_N$, $\boldsymbol{m}(p) \geq 1$, where $({}^\bullet x)_N := \{y | (y, x) \in \Phi\}$. For a given marking $\boldsymbol{m}$ the set of enabled transitions is denoted by the symbol $T_e(N, \boldsymbol{m})$. If $\Pi_1 \subseteq \Pi$ is a subset of places, then $\boldsymbol{m}|_{\Pi_1}$ denotes the projection of the marking $\boldsymbol{m}$ to places in Π_1. That is, $\forall p \in \Pi_1$, $\boldsymbol{m}|_{\Pi_1}(p) = \boldsymbol{m}(p)$. An enabled transition $t \in T_e(N, \boldsymbol{m})$ can *fire*, which changes the marking $\boldsymbol{m}^1$ to $\boldsymbol{m}^2$ according to

$$\boldsymbol{m}^2(p) = \boldsymbol{m}^1(p) - \mathrm{card}((p^\bullet)_N \cap \{t\}) + \mathrm{card}(({}^\bullet p)_N \cap \{t\}) \quad (1)$$

where the symbol $\mathrm{card}(\bullet)$ is used to denote the cardinality of the set argument, and $(x^\bullet)_N := \{y | (x, y) \in \Phi\}$. The set of input (output) places in the PN N to a subset of transitions $T_1 \subseteq T$ is denoted by the symbol $({}^\bullet T_1)_N$ ($(T_1^\bullet)_N$). A string of transitions $\sigma = t_{j_1} t_{j_2} \cdots t_{j_k}$, where $t_{j_i} \in T$ ($i \in \{1, 2, \cdots, k\}$) is said to be a *valid firing string* at the marking $\boldsymbol{m}$, if: 1) the transition t_{j_1} is enabled at the marking $\boldsymbol{m}$ and 2) for $i \in \{1, 2, \cdots, k-1\}$ the firing of the transition t_{j_i} produces a marking at which the transition $t_{j_{i+1}}$ is enabled.

Given an initial-marking $\boldsymbol{m}^0$ the set of *reachable markings* for $\boldsymbol{m}^0$ denoted by $\Re(N, \boldsymbol{m}^0)$ is the set of markings generated by all valid firing strings at the initial-marking $\boldsymbol{m}^0$ in the PN N. At a marking $\boldsymbol{m}^1$, if the firing of a valid firing string σ results in a marking $\boldsymbol{m}^2$, we represent it as $\boldsymbol{m}^1 \rightarrow \sigma \rightarrow \boldsymbol{m}^2$. A transition $t \in T$ is *live* if $\forall \boldsymbol{m}^1 \in \Re(N, \boldsymbol{m}^0)$, $\exists$ a $\boldsymbol{m}^2 \in \Re(N, \boldsymbol{m}^1)$ such that $t \in T_e(N, \boldsymbol{m}^2)$. The PN N is *live* if every transition $t \in T$ is live.

For any firing string $\sigma \in T^*$, we use the symbol $|\sigma|$ to denote the length of the string σ. If $T_1 \subseteq T$ is a subset of transitions, then $\sigma|_{T_1} \in T_1^*$ denotes the projection of the string σ to the alphabet T_1. That is, $\sigma|_{T_1}$ is the string of transitions in T_1 obtained from σ by erasing all transitions that are not in T_1, while retaining the transition order in the rest of the string.

The subnet, $N[T_i] = (\Pi_i, T_i, \Phi_i, \boldsymbol{m}_i^0)$, induced by a set of transitions $T_i \subseteq T$ is defined as $\Pi_i = ({}^\bullet T_i)_N \cup (T_i^\bullet)_N$, $\Phi_i = \Phi \cap ((\Pi_i \times T_i) \cup (T_i \times \Pi_i))$, and $\forall p \in \Pi_i$, $\boldsymbol{m}_i^0(p) = \boldsymbol{m}^0(p)$. Consider the PN shown in Fig. 1(a); the PN shown in Fig. 1(b) and (c) is the subnet $N[T_1]$ ($N[T_2]$) induced by the set of transitions $T_1 = \{t_1, t_2, t_3, t_4\}$ ($T_2 = \{t_5, t_6, t_7, t_8, t_9\}$). A place $p \in \Pi$ is said to be a *cut-place* for the PN N if the set of transitions T can be partitioned into two nonempty subsets $T_1, T_2 \subset T$ ($T_1 \neq \emptyset$, $T_2 \neq \emptyset$, $T_1 \cap T_2 = \emptyset$, and $T_1 \cup T_2 = T$) such that the subnets $N[T_1] = (\Pi_1, T_1, \Phi_1, \boldsymbol{m}_1^0)$ and $N[T_2] = (\Pi_2, T_2, \Phi_2, \boldsymbol{m}_2^0)$, have just the place p in common. That is, $\Pi_1 \cap \Pi_2 = \{p\}$. Additionally, we say the cut-place $p \in \Pi$ is a *directed cut-place* from $N[T_1]$ to $N[T_2]$ if $({}^\bullet p)_N \cap T_2 = \emptyset$. The place p_4 in the PN shown in Fig. 1(a) is a directed cut-place from the subnet $N[T_1]$ to the subnet $N[T_2]$, where the subsets T_1 and T_2 are defined above.

0018–9286/99$10.00 © 1999 IEEE

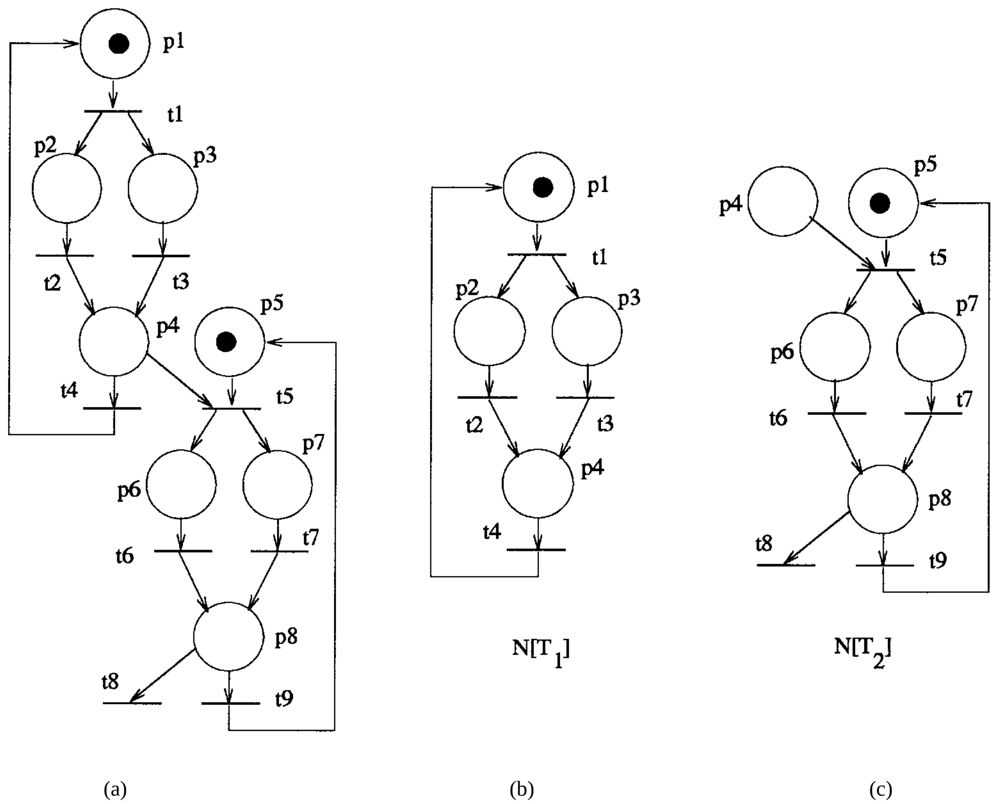

Fig. 1. Place p_4 is a directed cut-place from $N[T_1]$ to $N[T_2]$ for the PN shown in (a); the subnets induced by the set of transitions $T_1 = \{t_1, t_2, t_3, t_4\}$ and $T_2 = \{t_5, t_6, t_7, t_8, t_9\}$ are shown in (b) and (c), respectively.

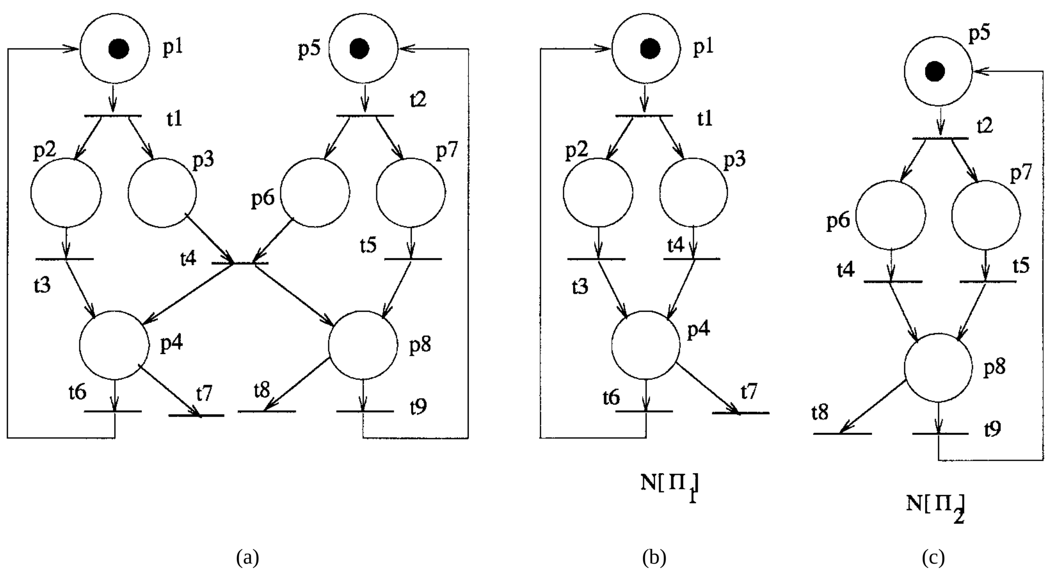

Fig. 2. Transition t_3 is a cut-transition of the PN shown in (a); the subnets induced by the set of places $\Pi_1 = \{p_1, p_3, p_5\}$ and $\Pi_2 = \{p_2, p_4, p_6\}$ are shown in (b) and (c), respectively.

Complementarily, the subnet, $N[\Pi_i] = (\Pi_i, T_i, \Phi_i, \boldsymbol{m}_i^0)$, induced by a set of places $\Pi_i \subseteq \Pi$, is defined as $T_i = ({}^\bullet\Pi_i)_N \cup (\Pi_i^\bullet)_N$, $\Phi_i = \Phi \cap ((\Pi_i \times T_i) \cup (T_i \times \Pi_i))$, and $\forall p \in \Pi_i$, $\boldsymbol{m}_i^0(p) = \boldsymbol{m}^0(p)$. Consider the PN shown in Fig. 2(a); the subnet induced by the set of places $\Pi_1 = \{p_1, p_3, p_5\}$ ($\Pi_2 = \{p_2, p_4, p_6\}$) is shown in Fig. 2(b) and (c). A transition $t \in T$ is said to be a *cut-transition* for the PN N if the set of places Π can be partitioned into two nonempty subsets $\Pi_1, \Pi_2 \subset \Pi$ ($\Pi_1 \neq \emptyset$, $\Pi_2 \neq \emptyset$, $\Pi_1 \cap \Pi_2 = \emptyset$, and $\Pi_1 \cup \Pi_2 = \Pi$) such that the subnets $N[\Pi_1] = (\Pi_1, T_1, \Phi_1, \boldsymbol{m}_1^0)$ and $N[\Pi_2] = (\Pi_2, T_2, \Phi_2, \boldsymbol{m}_2^0)$ have just the transition t in common. That is, $T_1 \cap T_2 = \{t\}$. For the PN shown in Fig. 2(a), the transition t_3 is a cut-transition.

A *completely controlled Petri net* (CCPN) [7] is expressed as an ordered six-tuple: $M = (\Pi, T, \Phi, \boldsymbol{m}^0, C, B)$, where $\Pi = \{p_1, p_2, \cdots, p_n\}$ is a set of n *state-places*, $T = \{t_1, t_2, \cdots, t_m\}$ is a set of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *state-arcs*; $C = \{c_1, c_2, \cdots, c_m\}$ is the set of *control-places*; $B = \{(c_i, t_i) | i = 1, 2, \cdots, m\}$ is the set of *control-arcs*; $\boldsymbol{m}^0 : \Pi \rightarrow \mathcal{N}$

is the *initial-marking function* (or the *initial-marking*), and $\mathcal{N}$ is the set of nonnegative integers. The CCPN $M = (\Pi, T, \Phi, \boldsymbol{m}^0, C, B)$ contains the underlying PN $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$. As there is one control place assigned to each transition the underlying PN uniquely determines the CCPN. Therefore, in graphical representations of CCPN's we do not explicitly represent the control places.

A *control* $\boldsymbol{u} : C \rightarrow \{0, 1\}$ assigns a token load of zero or one to each control place. The control can also be interpreted as an m-dimensional binary vector $\boldsymbol{u} \in \{0, 1\}^m$. It would help to view the control $\boldsymbol{u}$ as follows: if the ith component of $\boldsymbol{u}$, or $\boldsymbol{u}(c_i)$ is zero (one) then transition t_i is control-disabled (control-enabled). For a given marking $\boldsymbol{m}$ (control $\boldsymbol{u}$), a transition $t_i \in T$ is said to be state-enabled (control-enabled) if $t_i \in T_e(N, \boldsymbol{m})$ (if $\boldsymbol{u}(c_i) = 1$). A transition that is control-enabled and state-enabled can fire resulting in the marking given by (1). A *supervisory policy* $\mathcal{P} : \mathcal{N}^n \times T \rightarrow \{0, 1\}$ is a partial map that assigns a control for each reachable marking and each transition and is possibly undefined for the unreachable markings.

For a given CCPN and supervisory policy $\mathcal{P}$, a string of transitions $\sigma = t_{j_1} t_{j_2} \cdots t_{j_k}$, where $t_{j_i} \in T (i \in \{1, 2, \cdots, k\})$ is said to be a *valid firing string under supervision* at the marking $\boldsymbol{m}^1$, if: 1) the transition t_{j_1} is state-enabled at the marking $\boldsymbol{m}^1$, $\mathcal{P}(\boldsymbol{m}^1, t_{j_1}) = 1$ and 2) for $i \in \{1, 2, \cdots, k-1\}$ the firing of the transition t_{j_i} produces a marking $\boldsymbol{m}^i$ at which the transition $t_{j_{i+1}}$ is state-enabled and $\mathcal{P}(\boldsymbol{m}^i, t_{j_{i+1}}) = 1$.

For a given supervisory policy $\mathcal{P}$, the set of *reachable markings under supervision* for a CCPN M with initial-marking $\boldsymbol{m}^0$, denoted by $\Re(M, \boldsymbol{m}^0, \mathcal{P})$, is the set of markings generated by all valid firing strings under supervision at the marking $\boldsymbol{m}^0$ in the CCPN M. For the CCPN M, a transition $t_{j_i} \in T$ is *live* under $\mathcal{P}$ if $\forall \boldsymbol{m}^1 \in \Re(M, \boldsymbol{m}^0, \mathcal{P})$, $\exists$ a $\boldsymbol{m}^2 \in \Re(M, \boldsymbol{m}^1, \mathcal{P})$ such that $t_{j_i} \in T_e(N, \boldsymbol{m}^2)$ and $\mathcal{P}(\boldsymbol{m}^2, t_{j_i}) = 1$. A supervisory policy $\mathcal{P}$ enforces liveness in a CCPN M if all transitions in M are live under $\mathcal{P}$.

The following theorem from [7] characterizes supervisory policies that enforce liveness in an arbitrary PN where every transition can be prevented from firing by the supervisor.

Theorem 1 [7]: For a given PN $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$, there exists a supervisory policy $\mathcal{P} : \mathcal{N}^n \times T \rightarrow \{0, 1\}$ that enforces liveness, if and only if $\exists$ a valid firing string $\sigma = \sigma_1 \sigma_2$, in N, starting from $\boldsymbol{m}^0$, such that: 1) $\boldsymbol{m}^0 \rightarrow \sigma_1 \rightarrow \boldsymbol{m}^1 \rightarrow \sigma_2 \rightarrow \boldsymbol{m}^2$; 2) $\boldsymbol{m}^2 \geq \boldsymbol{m}^1$; and 3) all transitions appear at least once in the firing string σ_2.

Testing the requirement of Theorem 1 involves testing the nonemptiness of a real-valued feasible region defined by linear inequalities. This procedure has a time complexity that is polynomially related to the number of variables, which is equal to the number of vertices in the *coverability graph* (cf. [3, Sec. V-A], [4, Sec. IV-B-1]) of the underlying PN of the CCPN. However, the number of vertices in the coverability graph of a PN can be exponentially related to its size. Reference [9] concerns the application of the stepwise refinement/abstraction procedure of Suzuki and Murata [10] to alleviate the computational burden involved the synthesis of a supervisory policy that enforces liveness. Supervisory policies that enforce liveness in completely controlled PN's, whose underlying PN is a *free-choice Petri net* (FCPN) (cf. [6, Sec. VII-B]), are characterized in [8]. In particular, in [8], a class of CCPN's are identified for which a supervisory policy that enforces liveness is readily available (cf. [8, Sec. IV-A]). This class of FCPN's play a critical role in the two examples of the next section.

In the next section we consider the cases when the underlying PN of a CCPN M has: 1) a directed cut-place or 2) a cut-transition. We show in either case there is a supervisory policy that enforces liveness in the CCPN M if and only if there are similar policies for two CCPN's whose underlying PN's are related to the subnets involved in the definition of the directed cut-place, or, cut-transition. Additionally, we show that the supervisory policies that enforce liveness in these two CCPN's can be easily extended to a corresponding policy for the CCPN M. This procedure presents yet another "divide and conquer" technique, for the synthesis of supervisory policies that enforce liveness in CCPN's.

III. Main Results

Let $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$ be a connected PN, and let M be a CCPN that has N as its underlying PN. Additionally, let us suppose a place $\hat{p} \in \Pi$ is a directed cut-place from the subnet $N[T_1]$ to the subnet $N[T_2]$, where $T_1 \neq \emptyset$, $T_2 \neq \emptyset$, $T_1 \cap T_2 = \emptyset$, and $T_1 \cup T_2 = T$. For a $\hat{t} \notin T$, let $N_1 = (\Pi_1, T_1 \cup \{\hat{t}\}, \Phi_1, \boldsymbol{m}_1^0)$, where $\Pi_1 = (({}^\bullet T_1)_N \cup (T_1^\bullet)_N)$, $\Phi_1 = (\Phi \cap ((\Pi_1 \times T_1) \cup (T_1 \times \Pi_1))) \cup \{(\hat{p}, \hat{t})\}$, and $\forall p \in \Pi_1$, $\boldsymbol{m}_1^0(p) = \boldsymbol{m}^0(p)$. Also, let $N_2 = (\Pi_2, T_2, \Phi_2, \boldsymbol{m}_2^0)$, where $\Pi_2 = (({}^\bullet T_2)_N \cup (T_2^\bullet)_N) - \{\hat{p}\}$, $\Phi_2 = \Phi \cap ((\Pi_2 \times T_2) \cup (T_2 \times \Pi_2))$, and $\forall p \in \Pi_2$, $\boldsymbol{m}_2^0(p) = \boldsymbol{m}^0(p)$. That is, the PN N_1 is obtained from $N[T_1]$ by adding an extra arc from the directed cut-place $\hat{p}$ to the newly added transition $\hat{t}$, and the PN N_2 is obtained from the PN $N[T_2]$ by removing the directed cut-place $\hat{p}$ together with all arcs emanating from it.

For a given firing string $\sigma \in T^*$ in the PN N, we define corresponding firing strings in the PN's N_1 and N_2 using $\beta_1 : T^* \rightarrow \{T_1 \cup \{\hat{t}\}\}^*$ and $\beta_2 : T^* \rightarrow T_2^*$, where

$$\beta_1(\sigma t) = \begin{cases} \beta_1(\sigma)\hat{t}, & \text{if } t \in \hat{p}^\bullet \cap T_2 \\ \beta_1(\sigma)t, & \text{if } t \in T_1 \\ \beta_1(\sigma), & \text{otherwise} \end{cases}$$

and

$$\beta_2(\sigma t) = \begin{cases} \beta_2(\sigma)t, & \text{if } t \in T_2 \\ \beta_2(\sigma), & \text{otherwise} \end{cases}$$

where $\beta_1(\lambda) = \beta_2(\lambda) = \lambda$, and λ is the empty or null-string. The following observation relates valid firing strings in the PN N and corresponding valid firing strings in the PN's N_1 and N_2.

Observation 1: Let N, N_1, and N_2 be three PN's as defined above. If $\boldsymbol{m}^0 \rightarrow \sigma \rightarrow \boldsymbol{m}^1$ in the PN N, then: 1) $\boldsymbol{m}^0|_{\Pi_1} \rightarrow \beta_1(\sigma) \rightarrow \boldsymbol{m}^1|_{\Pi_1}$ in N_1 and 2) $\boldsymbol{m}^0|_{\Pi_2} \rightarrow \beta_2(\sigma) \rightarrow \boldsymbol{m}^1|_{\Pi_2}$ in N_2.

Proof: This observation is established via an induction argument over $|\sigma|$, the length of the string σ. The base case is easily established by letting σ equal the null string, λ. As the induction hypothesis let us assume the statement of the observation is true of any σ, such that $|\sigma| \leq n$, for some $n \in \mathcal{N}$. For the induction step let $\boldsymbol{m}^0 \rightarrow \sigma \rightarrow \boldsymbol{m}^1 \rightarrow t \rightarrow \boldsymbol{m}^2$ in the PN N, for some $t \in T$. By the induction hypothesis we know $\boldsymbol{m}^0|_{\Pi_1} \rightarrow \beta_1(\sigma) \rightarrow \boldsymbol{m}^1|_{\Pi_1}$ in the PN N_1, and $\boldsymbol{m}^0|_{\Pi_2} \rightarrow \beta_2(\sigma) \rightarrow \boldsymbol{m}^1|_{\Pi_2}$ in the PN N_2.

Case 1—$t \in T_1$: Since $\boldsymbol{m}^1 \rightarrow t \rightarrow \boldsymbol{m}^2$ in N, it follows that $\forall p \in ({}^\bullet t)_N$, $\boldsymbol{m}^1(p) \neq 0$. Since $({}^\bullet t)_{N_1} = ({}^\bullet t)_N$, from the definition of $\boldsymbol{m}^1|_{\Pi_1}$, we infer $\forall p \in ({}^\bullet t)_{N_1}$, $\boldsymbol{m}^1|_{\Pi_1}(p) \neq 0$. Since $(t^\bullet)_{N_1} = (t^\bullet)_N$, from the definition of $\boldsymbol{m}^2|_{\Pi_1}$, we infer $\boldsymbol{m}^1|_{\Pi_1} \rightarrow \beta_1(t) \rightarrow \boldsymbol{m}^2|_{\Pi_1}$, as $\beta_1(t) = t$. Since $t \in T_1$, it trivially follows that $\boldsymbol{m}^1|_{\Pi_2} \rightarrow \beta_2(t) \rightarrow \boldsymbol{m}^2|_{\Pi_2}$, $\boldsymbol{m}^1|_{\Pi_2} = \boldsymbol{m}^2|_{\Pi_2}$, as $\beta_2(t) = \lambda$.

Case 2—$t \in (T_2 - (\hat{p}^\bullet)_N)$: It follows that $({}^\bullet t)_{N_2} = ({}^\bullet t)_N$, and $(t^\bullet)_{N_2} = (t^\bullet)_N$. Using an argument similar to that presented above, it can be shown that if $\boldsymbol{m}^1 \rightarrow t \rightarrow \boldsymbol{m}^2$ in N then: 1) $\boldsymbol{m}^1|_{\Pi_1} \rightarrow \beta_1(t) \rightarrow \boldsymbol{m}^2|_{\Pi_1}$ (in fact, $\boldsymbol{m}^1|_{\Pi_1} = \boldsymbol{m}^2|_{\Pi_2}$, $\beta_1(t) = \lambda$) and 2) $\boldsymbol{m}^1|_{\Pi_2} \rightarrow \beta_2(t) \rightarrow \boldsymbol{m}^2|_{\Pi_2}$. The repetition of the argument is skipped in the interest of space.

Case 3—$t \in (T_2 \cap (\hat{p}^\bullet)_N)$: From the fact that $\hat{p}$ is a directed cut-place, and $t \in T_2 \cap (\hat{p}^\bullet)_N$, we infer $(t^\bullet)_N \cap \Pi_1 = \emptyset$, $({}^\bullet t)_N \cap \Pi_1 = \{\hat{p}\}$ and $(t^\bullet)_N \subseteq \Pi_2$. So: 1) $\forall p \in (\Pi_1 - \{\hat{p}\})$, $\boldsymbol{m}^2(p) = \boldsymbol{m}^1(p)$; 2) $\boldsymbol{m}^2(\hat{p}) = \boldsymbol{m}^1(\hat{p}) - 1$, where $\boldsymbol{m}^1(\hat{p}) \neq 0$; and 3) $\forall p \in \Pi_2 - \{\hat{p}\}$, $\boldsymbol{m}^2(p) = \boldsymbol{m}^1(p) - \mathrm{card}((p^\bullet)_N \cap \{t\}) + \mathrm{card}(({}^\bullet p)_N \cap \{t\})$. Noting that 1) $\forall p \in \Pi_2 - \{\hat{p}\}$, $({}^\bullet p)_{N_2} = ({}^\bullet p)_N$ and $(p^\bullet)_{N_2} = (p^\bullet)_N$, and 2) $\forall t \in T_2$, $({}^\bullet t)_{N_2} = ({}^\bullet t)_N - \{\hat{p}\}$, and $(t^\bullet)_{N_2} = (t^\bullet)_N$, we note that $\boldsymbol{m}^1|_{\Pi_2} \to t \to \boldsymbol{m}^2|_{\Pi_2}$ in N_2 and $\boldsymbol{m}^1|_{\Pi_1} \to \hat{t} \to \boldsymbol{m}^2|_{\Pi_1}$ in N_1, hence the result. □

We now state and prove one of the two main results of the paper.

Theorem 2: Let M be a CCPN whose underlying PN is $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$, where $\hat{p} \in \Pi$ is a directed cut-place from $N[T_1]$ to $N[T_2]$, for a partition $T_1, T_2 \subseteq T$. Also, let M_1 and M_2 be two other CCPN's whose underlying PN's are N_1 and N_2, where the PN's N_1 and N_2 are constructed from $N[T_1]$ and $N[T_2]$ using the procedure defined above. There exists a supervisory policy that enforces liveness in M if and only if there are corresponding policies for the CCPN's M_1 and M_2.

Proof (Only If): Let us suppose there is a supervisory policy that enforces liveness in the CCPN M, then from Theorem 2 we conclude that $\exists \sigma_1, \sigma_2 \in T^*, \exists \boldsymbol{m}^1, \boldsymbol{m}^2$, such that $\boldsymbol{m}^0 \to \sigma_1 \to \boldsymbol{m}^1 \to \sigma_2 \to \boldsymbol{m}^2$ in $N, \boldsymbol{m}^2 \geq \boldsymbol{m}^1$, and all transitions appear at least once in σ_2. From Observation 1 we know that: 1) $\boldsymbol{m}^0|_{\Pi_1} \to \beta_1(\sigma_1) \to \boldsymbol{m}^1|_{\Pi_1} \to \beta_1(\sigma_2) \to \boldsymbol{m}^2|_{\Pi_1}$ in N_1 and 2) $\boldsymbol{m}^0|_{\Pi_2} \to \beta_2(\sigma_1) \to \boldsymbol{m}^1|_{\Pi_2} \to \beta_2(\sigma_2) \to \boldsymbol{m}^2|_{\Pi_2}$ in N_2. Since all transitions in T appear at least once in σ_2, it follows from the definition of $\beta_1(\bullet)$ ($\beta_2(\bullet)$) that all transitions in $T_1 \cup \{\hat{t}\}$ (T_2) appear at least once in $\beta_1(\sigma_2)$ ($\beta_2(\sigma_2)$). Additionally, from the fact that $\boldsymbol{m}^2 \geq \boldsymbol{m}^1$ we infer $\boldsymbol{m}^2|_{\Pi_1} \geq \boldsymbol{m}^1|_{\Pi_1}$ and $\boldsymbol{m}^2|_{\Pi_2} \geq \boldsymbol{m}^1|_{\Pi_2}$. From Theorem 2 we conclude there are supervisory policies that enforce liveness in the CCPN's M_1 and M_2.

(If): Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be supervisory policies that enforce liveness in the CCPN's M_1 and M_2, respectively. We construct a supervisory policy $\mathcal{P}$ for the CCPN M as follows:

$$\mathcal{P}(\boldsymbol{m}, t) = \begin{cases} \mathcal{P}_1(\boldsymbol{m}|_{\Pi_1}, t), & \text{if } t \in T_1 \\ \mathcal{P}_1(\boldsymbol{m}|_{\Pi_1}, \hat{t}) \wedge \mathcal{P}_2(\boldsymbol{m}|_{\Pi_2}, t), & \text{if } t \in T_2 \cap (\hat{p}^\bullet)_N \\ \mathcal{P}_2(\boldsymbol{m}|_{\Pi_2}, t), & \text{if } t \in T_2 - (\hat{p}^\bullet)_N. \end{cases}$$

We now show that the supervisory policy $\mathcal{P}$ enforces liveness in M. Let $\boldsymbol{m}^0 \to \sigma \to \boldsymbol{m}^1$ under the supervision of $\mathcal{P}$ in M. Using an induction argument along the lines of a proof of Observation 1, we can establish that $\boldsymbol{m}^0|_{\Pi_1} \to \beta_1(\sigma) \to \boldsymbol{m}^1|_{\Pi_1}$ in M_1 under the supervision of $\mathcal{P}_1$, and $\boldsymbol{m}^0|_{\Pi_2} \to \beta_2(\sigma) \to \boldsymbol{m}^1|_{\Pi_2}$ in M_2 under the supervision of $\mathcal{P}_2$. The details are skipped in the interest of space.

Since $\mathcal{P}_1$ enforces liveness in M_1 it follows that $\forall t \in T_1, \exists \sigma_1 \in T_1^*, \exists \boldsymbol{m}_1^2 \in \mathcal{N}^{\text{card}(\Pi_1)}$, such that $\boldsymbol{m}^1|_{\Pi_1} \to \sigma_1 \to \boldsymbol{m}_1^2$ under the supervision of $\mathcal{P}_1$ in M_1, and the transition t is control- and state-enabled at the marking $\boldsymbol{m}_1^2$. Using an induction argument over the length of σ_1, it can be shown that $\boldsymbol{m}^1 \to \sigma_1 \to \boldsymbol{m}^2$ under the supervision of $\mathcal{P}$ in M, where $\boldsymbol{m}^2|_{\Pi_1} = \boldsymbol{m}_1^2$. The base case of this induction argument is easily established by letting σ_1 be null or empty string. Assuming the above statement to be true of any $\sigma_1 \leq n$ for some $n \in \mathcal{N}$, the induction step can be established by noting that $\forall t \in T_1, ({}^\bullet t)_{N_1} = ({}^\bullet t)_N$ and $(t^\bullet)_{N_1} = (t^\bullet)_N$, the fact that $\forall \boldsymbol{m} \in \mathcal{N}^{\text{card}(\Pi)}, \mathcal{P}(\boldsymbol{m}, t) = \mathcal{P}_1(\boldsymbol{m}|_{\Pi_1}, t)$. The details of this routine induction proof are skipped for brevity. Since the transition t is state- and control-enabled at the marking $\boldsymbol{m}_1^2$ in M_1, and since $({}^\bullet t)_N = ({}^\bullet t)_{N_1}$, it follows that t is state-enabled at the marking $\boldsymbol{m}^2$ in M. From the definition of $\mathcal{P}$ it follows that t is also control-enabled at the marking $\boldsymbol{m}^2$ in M under $\mathcal{P}$. It is important to note that none of the transitions in T_2 appear in the string σ_1. Hence all transitions in T_1 are live under the supervisory policy $\mathcal{P}$.

Since $\mathcal{P}_1$ enforces liveness in M_1 and the transition $\hat{t}$ is live under its supervision in M_1, and since $({}^\bullet\hat{t})_{N_1} = \{\hat{p}\}$ and $(\hat{t}^\bullet)_{N_1} = \emptyset$, it follows that the token-load of the directed cut-place $\hat{p}$ can be made arbitrarily large by just firing transitions in T_1 alone in M_1. To see this, we note that every instance of the firing of $\hat{t}$ in M_1 can be replaced by a scenario where the $\hat{t}$ is *not* fired, and instead the token that would have been consumed by the firing of $\hat{t}$ is retained forever in $\hat{p}$. Using the same argument as above, it can be shown that the token load of $\hat{p}$ in M can be made arbitrarily large under the supervision of $\mathcal{P}$.

The argument that establishes the liveness of transitions in T_1 under $\mathcal{P}$ in M cannot be repeated *in toto* for the case when $t \in T_2$; this is because $\forall t \in T_2, ({}^\bullet t)_{N_2} = ({}^\bullet t)_N - \{\hat{p}\}, (t^\bullet)_{N_2} = (t^\bullet)_N$. In particular, if $t \in (\hat{p}^\bullet)_N$, since $\hat{p} \notin ({}^\bullet t)_{N_2}$ and $\hat{p} \in ({}^\bullet t)_N$, the existence of a firing string in M_2 that is valid under the supervision of $\mathcal{P}_2$ does not directly imply that a corresponding string exists in M under the supervision of $\mathcal{P}$. The existence of the corresponding firing string can be inferred only after guaranteeing $\hat{p}$ has sufficient tokens for the firing of each member of $(\hat{p}^\bullet)_N$. Since the token load of $\hat{p}$ in M can be made arbitrarily large under the supervision of $\mathcal{P}$ by firing only transitions that belong to T_1, this guarantee can be made. The routine, yet laborious details of an induction argument that establishes this claim are skipped for brevity. Essentially, we conclude that for any $t \in T_2, \exists \sigma_2 \in T^*, \exists \boldsymbol{m}^2 \in \mathcal{N}^{\text{card}(\Pi)}$, such that $\boldsymbol{m}^1 \to \sigma_2 \to \boldsymbol{m}^2$ in M and such that t is control- and state-enabled at $\boldsymbol{m}^2$ in M under the supervision of $\mathcal{P}$, hence the result. □

We now turn our attention to the case when the underlying PN of a CCPN has a cut-transition. Let M be a CCPN with an underlying connected PN $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$. Let $\hat{t} \in T$ be a cut-transition for N for a partition $\Pi_1, \Pi_2 \subseteq \Pi$. Also, let $N_1 = N[\Pi_1]$ and $N_2 = N[\Pi_2]$, where $N_i = (\Pi_i, T_i, \Phi_i, \boldsymbol{m}_i^0)$ $(i = 1, 2)$. Essentially, the PN N is obtained by merging the PN's N_1 and N_2 at the cut-transition $\hat{t}$. This fact results in Observation 2, which is used in the proof of the other main result of this paper Theorem 3. The proof of Observation 2 is straightforward and is skipped in the interest of space.

Observation 2: Let N, N_1, and N_2 be three PN's as defined above. If $\boldsymbol{m}^0 \to \sigma \to \boldsymbol{m}^1$ in the PN N, then: 1) $\boldsymbol{m}^0|_{\Pi_1} \to \sigma|_{T_1} \to \boldsymbol{m}^1|_{\Pi_1}$ in N_1 and 2) $\boldsymbol{m}^0|_{\Pi_2} \to \sigma|_{T_2} \to \boldsymbol{m}^1|_{\Pi_1}$ in N_2.

Theorem 3: Let M be a CCPN whose underlying PN is $N = (\Pi, T, \Phi, \boldsymbol{m}^0)$, where $\hat{t} \in T$ is a cut-transition for a partition $\Pi_1, \Pi_2 \subseteq \Pi$. Also, let M_1 and M_2 be two other CCPN's whose underlying PN's are $N_1 (= N[\Pi_1])$ and $N_2 (= N[\Pi_2])$, where $N_i = (\Pi_i, T_i, \Phi_i, \boldsymbol{m}_i^0)(i = 1, 2)$. There exists a supervisory policy that enforces liveness in M if and only if there are corresponding policies for the CCPN's M_1 and M_2.

Proof (Only If): This part of the proof follows directly from Observation 2 and an argument that is similar to the proof of the only-if part of Theorem 3. We refrain from presenting it again in the interest of space.

(If): Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be supervisory policies that enforce liveness in the CCPN's M_1 and M_2, respectively. We construct a supervisory policy $\mathcal{P}$ for the CCPN M as follows:

$$\mathcal{P}(\boldsymbol{m}, t) = \begin{cases} \mathcal{P}_1(\boldsymbol{m}|_{\Pi_1}, t), & \text{if } t \in (T_1 - \{\hat{t}\}) \\ \mathcal{P}_2(\boldsymbol{m}|_{\Pi_2}, t), & \text{if } t \in (T_2 - \{\hat{t}\}) \\ \mathcal{P}_2(\boldsymbol{m}|_{\Pi_2}, t) \wedge \mathcal{P}_2(\boldsymbol{m}|_{\Pi_2}, t), & \text{if } t = \{\hat{t}\}. \end{cases}$$

We now show that the supervisory policy $\mathcal{P}$ enforces liveness in M. Let $\boldsymbol{m}^0 \to \sigma \to \boldsymbol{m}^1$ under the supervision of $\mathcal{P}$ in M. Using an induction argument on the length of σ it can be shown that: 1) $\boldsymbol{m}^0|_{\Pi_1} \to \sigma|_{T_1} \to \boldsymbol{m}^1|_{\Pi_1}$ under the supervision of $\mathcal{P}_1$ in M_1 and 2) $\boldsymbol{m}^0|_{\Pi_2} \to \sigma|_{T_2} \to \boldsymbol{m}^1|_{\Pi_2}$ under the supervision of $\mathcal{P}_2$ in M_2. The details of this induction argument are skipped for brevity.

Since $\mathcal{P}_1$ enforces liveness in M_1, we infer $\forall t \in T_1, \exists \sigma_1 \in T_1^*, \exists \boldsymbol{m}_1^2 \in \mathcal{N}^{\text{card}(\Pi_1)}$, such that $\boldsymbol{m}^1|_{\Pi_1} \to \sigma_1 \to \boldsymbol{m}_1^2$ under the supervision of $\mathcal{P}_1$ in M_1 and t is control- and state-enabled at the marking $\boldsymbol{m}_1^2$ in M_1. Let $\sigma_1 = \hat{\sigma}_1 \hat{t} \hat{\sigma}_2 \hat{t} \cdots \hat{t} \hat{\sigma}_k$, where $\hat{\sigma}_i \in (T_1 - \{\hat{t}\})^*$. Using an induction argument over k, the number of occurrences of the cut-transition $\hat{t}$ in σ_1 it can be shown that appropriate firing strings $\tilde{\sigma}_i \in (T_2 - \{\hat{t}\})^*, i = 1, \cdots, k$ can be intercalated into σ_1 to create a firing string $\hat{\sigma} = \hat{\sigma}_1 \tilde{\sigma}_1 \hat{t} \hat{\sigma}_2 \tilde{\sigma}_2 \hat{t} \cdots \tilde{\sigma}_{k-1} \hat{t} \hat{\sigma}_k \tilde{\sigma}_k$ that is valid under the supervision of $\mathcal{P}$ in M at the marking $\boldsymbol{m}^1$. A critical component of the induction argument is that the policy $\mathcal{P}_2$ enforces liveness in N_2 and $T_1 \cap T_2 = \hat{t}$. The details of this routine, yet laborious, process are skipped in the interest of space. Additionally, if $\boldsymbol{m}^1 \rightarrow \hat{\sigma} \rightarrow \boldsymbol{m}^2$ under the supervision of $\mathcal{P}$ in M, it can be shown that $\boldsymbol{m}^2|_{\Pi_1} = \boldsymbol{m}_1^2$. So, from the definition of $\mathcal{P}$, if $t \in (T_2 - \{\hat{t}\})$, t is state- and control-enabled under $\mathcal{P}$ at the marking $\boldsymbol{m}^2$. If $t = \hat{t}$, then by the appropriate choice $\tilde{\sigma}_k$ it can be shown that $\hat{t}$ is both control- and state enabled under $\boldsymbol{m}^2$. Using a similar argument the liveness of any transition in $T_2 - \hat{t}$ can also be established. Hence the result. □

We illustrate the utility of Theorem 2. Consider a CCPN M whose underlying PN is the PN shown in Fig. 1(a). The PN's N_1 and N_2 constructed as per the procedure outline before the statement of Theorem 2 are essentially similar to the PN $N[T_2]$ where the place p_4 and the arc from place p_4 to transition t_5 is removed. The coverability graph of the PN N has 700 nodes, while that of N_1 or N_2 has 25 nodes. The PN's N_1 and N_2 belong to a class of PN's for which a supervisory policy that enforces liveness is readily available (cf. [8, Sec. IV-A]). A supervisory policy that guarantees the nonemptiness of the place set $\{p_1, p_2, p_3, p_4\}$ ($\{p_5, p_6, p_7, p_8\}$) enforces liveness in N_1 (N_2). Theorem 2 shows that the policy that simultaneously guarantees the nonemptiness of the place sets $\{p_1, p_2, p_3, p_4\}$ and $\{p_5, p_6, p_7, p_8\}$ enforces liveness in M.

Consider a CCPN M whose underlying PN N is shown in Fig. 2(a). The underlying PN's N_1 and N_2 for the CCPN's M_1 and M_2 are shown in Fig. 2(b) and (c). The coverability graph of N has 300 nodes, while that of the PN's N_1 and N_2 have 13 nodes. Additionally, the PN's in Fig. 2(b) and (c) also belong to the class of PN's for which a supervisory policy that enforces liveness is readily available (cf. [8, Sec. IV-A]). The policy that enforces the nonemptiness of the place set $\{p_1, p_3, p_5\}$ ($\{p_2, p_4, p_6\}$) enforces liveness in N_1 (N_2). The policy for M constructed using the procedure outlined in the proof of Theorem 3 enforces the nonemptiness of $\{p_1, p_3, p_5\}$ and $\{p_2, p_4, p_6\}$. This policy enforces liveness in M.

In the general case, the complexity of synthesizing a supervisory policy that enforces liveness in a CCPN M involves a procedure on the coverability graph of its underlying PN N. However, when the structure of N has either a directed cut-place or a cut-transition, the results of this paper show that the synthesis of the supervisory policy for M is equivalent to the synthesis of similar policies for two CCPN's M_1 and M_2, where N_1 (N_2) is the underlying PN of the CCPN M_1 (M_2).

By interpreting places and transitions in a PN as vertices in a graph, the connectedness of an arbitrary PN can be tested efficiently (cf. [2, Algorithm 1, Sec. XI-D]). Additionally, there are efficient procedures for testing the existence of cut-vertices in an arbitrary graph (cf. [2, Algorithm 4, Sec. XI-D]). These procedures can be directly used to investigate the existence of cut-places and cut-transitions in an arbitrary PN. If a cut-place is found, testing if this cut-place is a directed cut-place is also straightforward.

IV. Conclusions

Testing the existence and the synthesis of supervisory policies that enforce liveness in connected PN's [4], [6] is computationally burdensome, in general [7]. In this paper we presented a "divide-and-conquer" approach that alleviates this computational burden when the plant PN has either directed cut-places or cut-transitions. A place (transition) is said to be a cut-place (cut-transition) if its removal from the PN would result in two separate PN's. A cut-place is said to be a directed cut-place if in the original PN all its input transitions originate from only one of the two components that were created when the cut-place is removed. The computational benefit to this approach was illustrated by two examples.

References

[1] M. Ben-Ari, *Principles of Concurrent Programming.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[2] N. Deo, *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

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

[4] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

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

[6] W. Reisig, *Petri Nets.* Berlin, Germany: Springer-Verlag, 1985.

[7] R. S. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete event dynamic systems modeled by controlled Petri nets," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 928–945, July 1996.

[8] ——, "On commoner's liveness theorem and supervisory policies that enforce liveness in free-choice Petri nets," *Syst. Contr. Lett.*, vol. 31, pp. 41–48, 1997.

[9] ——, "On supervisory policies that enforce liveness in a class of completely controlled Petri nets obtained via refinement," *IEEE Trans. Automat. Contr.*, 1997.

[10] I. Suzuki and T. Murata, "A method for stepwise refinement and abstraction of Petri nets," *J. Computer and Syst. Sci.*, vol. 27, pp. 51–76, 1983.

Interpolation of Observer State Feedback Controllers for Gain Scheduling

Daniel J. Stilwell and Wilson J. Rugh

***Abstract*—The authors propose a method of interpolating linear time-invariant controllers with observer state feedback structure in order to generate a continuously varying family of controllers that stabilizes a family of linear plants. Gain scheduling is a motivation for this work, and the interpolation method yields guidelines for the design of gain scheduled controllers. The method is illustrated with the design of a missile autopilot using loop-shaping H-infinity controllers.**

***Index Terms*—Gain scheduling, linear parameter-varying.**

I. Introduction

To motivate the controller interpolation issue, we briefly describe a typical gain scheduling formulation. Consider a nonlinear plant of the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \qquad t \geq 0 \\ y(t) &= h(x(t))) \end{aligned} \tag{1}$$

Manuscript received September 17, 1997. Recommended by Associate Editor, A. Rantzer. This work was supported by the Air Force Office of Scientific Research.

D. J. Stilwell is with the Systems Engineering Department, U.S. Naval Academy, Annapolis, MD 21402 USA.

W. J. Rugh is with the Department of Electrical & Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: rugh@jhu.edu).

Publisher Item Identifier S 0018-9286(99)03939-2.