Non-blocking Supervisory Control of Nondeterministic Systems via Prioritized Synchronization [1]

Ratnesh Kumar
Department of Electrical Engineering
University of Kentucky
Lexington, KY 40506-0046
Email: kumar@engr.uky.edu

Mark A. Shayman
Department of Electrical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742
Email: shayman@eng.umd.edu

November 20, 2005

[1]This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, in part by by the National Science Foundation under the Grants NSFD-CDR-8803012, NSF-ECS-9409712, NSF-ECS-9312587, the Minta Martin Fund for Aeronautical Research, and the General Research Board at the University of Maryland.

Abstract

In a previous paper we showed that supervisory control of nondeterministic discrete event systems, in the presence of driven events, can be achieved using prioritized synchronous composition as a mechanism of control, and trajectory models as a modeling formalism, first introduced by Heymann. The specifications considered in this earlier work were given by *prefix-closed* languages. In this paper, we extend this work to include markings so that non-closed specifications and issues such as blocking can be addressed. It is shown that the usual notion of non-blocking, called *language model non-blocking*, may not be adequate in the setting of nondeterministic systems, and a stronger notion, called *trajectory model non-blocking*, is introduced. Necessary and sufficient conditions for the existence of language model non-blocking as well as trajectory model non-blocking supervisors are obtained for nondeterministic systems in the presence of driven events in terms of extended controllability and relative-closure conditions, and a new condition called the trajectory-closure condition.

Keywords: discrete event systems, supervisory control, nondeterministic automata, driven events, prioritized synchronization, trajectory models, blocking

1 Introduction

Discrete event systems involve quantities which take a discrete set of values which remain constant except at discrete times when events occur in the system. Examples include communication networks, intelligent vehicle highway systems, manufacturing systems and computer programs. Supervisory control theory was developed to provide a mathematical framework for the design of controllers for such systems in order to meet various *qualitative* constraints. For more details refer to [15, 17, 10].

The majority of the research effort in this area has focused on the supervisory control of *deterministic* systems, and relatively little progress has been made towards that of *nondeterministic* systems–systems in which knowledge of the current state and next event is insufficient to uniquely determine the next state. Such nondeterminism arises due to unmodeled system dynamics and/or partial observation. For example a change-giving machine may give a different combination of coins as change (for the same input amount) depending on the sequence in which coins are loaded in the machine. However, for simplicity, this detail may be suppressed while obtaining a model for the machine leading to a nondeterministic model of it. Similarly, a machine in a manufacturing system may incur a partial undetectable failure while performing a certain task. This can be modeled by having a nondeterministic transition on the task completion event leading to two successor states depending on whether or not the failure occurred while completing the task. Also, in a communication network, a user is only able to observe the *external* events such as transmission and reception of messages, whereas the *internal* events such as loss or collision of messages, acknowledgments, etc., are not observed. Such internal events can be represented as silent or ϵ-*transitions* leading to a nondeterministic model of the communication network.

In the Ramadge-Wonham approach to supervisory control, every event is generated by the plant and synchronously executed by the supervisor [14] which acts passively by disabling certain controllable events possible in the open-loop plant. The disablement action is accomplished by a control-input map which specifies a set of disabled events based on the current state of the supervisor. Alternatively, in the work of Kumar-Garg-Marcus [11], the disablement action is accomplished by removing certain transitions from the structure of the supervisor while continuing to require that the plant and supervisor be connected by strict synchronous composition (SSC). In the work of Golaszewski-Ramadge [3] and, in the real-time setting, the work of Brandin-Wonham [2], the supervisor is able to initiate certain so-called *forcible* events that the plant synchronously executes. In the work of Balemi and coworkers [1], events can originate in the supervisor (so-called *command events*) or in the plant (so-called *response events*). The assumption is made that the plant and supervisor are *mutually receptive*, meaning that neither the plant nor the supervisor can refuse to execute an event initiated by the other.

Common to all of the above approaches is the assumption that there are never events which may occur in the supervisor without the participation of the plant. However, this assumption may be unreasonably restrictive for nondeterministic systems. When the plant is nondeterministic, there is generally no way to know a priori whether a command issued

by the supervisor can be executed by the plant in its current state. For example, it may be impossible to know that a device is in a faulted state until after it fails to respond to a command from the controller.

Heymann has introduced an interconnection operator called *prioritized synchronous composition* (PSC) [4], which relaxes the synchronization requirements between the plant and supervisor. Each process in a PSC-interconnection is assigned a *priority set* of events. For an event to be enabled in the interconnected system, it must be enabled in all processes whose priority sets contain that event. Also, when an enabled event occurs, it occurs in each subsystem in which the event is enabled. In the context of supervisory control, the priority set of the plant contains the controllable and uncontrollable events, while the priority set of the supervisor contains the controllable and driven events. Thus, controllable events require the participation of both plant and supervisor; uncontrollable events require the participation of the plant and will occur synchronously in the supervisor whenever possible; driven events require the participation of the supervisor and will occur synchronously in the plant whenever possible.

It is important to distinguish between PSC and other types of parallel composition in the literature. For example, Hoare [7] defines a concurrent composition operator in which each process has its own alphabet and the processes synchronize on the events in the intersection of their alphabets. This is generalized to trace-dependent alphabets, called event-control sets, by Inan-Varaiya [9]. The key difference between concurrent composition and PSC is that in PSC, although a process cannot block events which are outside its priority set, it may be able to execute these events–and, whenever possible, will execute these events synchronously when they occur in the other process[1].

Language models identify processes that have the same set of traces. The failures model of Hoare [7] identifies processes that have the same set of so-called *failures*. Failure equivalence refines language equivalence. Heymann showed that failure equivalence is too coarse to support the PSC operator [4]. In other words, there exist two different plants with the same failures model (and hence with the same language model) such that their PSC's with a common supervisor have different language models. Thus, neither the language model nor even the failures model retains enough information about a process to do control design using the operation of PSC.

This has led Heymann to introduce the *trajectory model*, a refinement of the failures model [4, 6]. The trajectory model is similar to the *failure-trace model* (also called the refusal-testing model) in concurrency theory [13], but differs from this model in its treatment of hidden transitions. The trajectory model treats hidden transitions in a way that is consistent with the failures model. In a previous paper [16], we proved that the trajectory model retains sufficient process detail to permit PSC-based controller design.

In [16], we showed that supervisory control of nondeterministic discrete event systems, in

[1]If applied to so-called *improper* processes, the parallel operator defined by Inan [8] can be viewed as a generalized form of PSC, but only in the deterministic setting. However, when supervisory control is considered in this reference, the assumption is made that the plant is proper and has a constant event control set. This assumption excludes driven events.

the presence of driven events, can be achieved using prioritized synchronous composition as a mechanism of control, and trajectory models as a modeling formalism. The specifications considered in [16] were given by *prefix-closed* languages. In this paper, we extend our earlier work to include the notion of markings by introducing the notion of recognized and generated trajectory sets, so that non-closed specifications and the issue of blocking can be addressed.

It is important to understand the difference between the "absence of deadlock" and the "absence of blocking". While the former can be described using refusal sets, the later needs the notion of marking: The absence of deadlock requires that the controlled system should never reach a state with refusal set being the entire event set, whereas the absence of blocking requires that the controlled system should never reach a state from where it is not possible to *complete* its execution by reaching a marked state. If the refusal set of a state is the entire event set, then this means that no events are possible in that state and the system would deadlock. However, that state may represent completion of a task, in which case the state would be marked and not result in blocking. On the other hand, the system could become trapped in a cycle that contains no states that represent completion of tasks–i.e., no marked states. In this case, the refusal set would not be the entire event set, i.e., system would not deadlock, and yet the system is blocked since it is in livelock. Consequently, refusal sets are insufficient to distinguish blocking and nonblocking states, and the notion of marking is required.

The usual notion of non-blocking, referred to as *language model non-blocking* in this paper, requires that each trace belonging to the generated language of a controlled system be extendable to a trace belonging to the recognized language. This property adequately captures the notion of non-blocking in a deterministic setting. However, in a nondeterministic setting, the execution of a certain trace belonging to the generated behavior may lead to more than one state. Language model non-blocking only requires that each such trace be extendable to a trace in the recognized behavior from *at least one* such state–as opposed to *all* such states. Thus, a language model non-blocking nondeterministic system can get blocked, as illustrated by the example Section 6. Consequently, there is a need for a stronger type of non-blocking for nondeterministic systems. This leads us to introduce the property of *trajectory model non-blocking*, which requires that each refusal-trace belonging to the generated trajectory set of a nondeterministic system be extendable to a refusal-trace belonging to the recognized trajectory set. This stronger notion of non-blocking seems adequate for practical systems, although as explained in Remark 4 it does not always guarantee the absence of "blocking".

Another desirable property of a supervisor is that it should be *non-marking*, i.e., a certain trace (respectively, a refusal-trace) of the controlled system should belong to the recognized language (respectively, the recognized trajectory set) of the controlled system if and only if a marked state of the uncontrolled system is reached due to its execution regardless of the type of state reached in the supervisor. We first obtain a necessary and sufficient condition for the existence of a non-marking and language model non-blocking supervisor for a given nondeterministic system in the presence of driven events. This result is then used to obtain a necessary and sufficient condition for the existence of a non-marking and trajectory model

non-blocking supervisor in that setting.

2 Notation and Preliminaries

Given a finite event set Σ, Σ^* is used to denote the collection of all *traces*, i.e., finite sequences of events, including the zero length sequence, denoted by ϵ. A subset of Σ^* is called a language. Symbols H, K, etc. are used to denote languages. The set $2^\Sigma(\Sigma \times 2^\Sigma)^*$ is used to denote the collection of all *refusal-traces*, i.e., finite sequences of alternating *refusals* and events [6, 16] of the type:

$$\Sigma_0(\sigma_1, \Sigma_1) \dots (\sigma_n, \Sigma_n),$$

where $n \in \mathcal{N}$. The sequence $\sigma_1 \dots \sigma_n \in \Sigma^*$ is the trace, and for each $i \leq n$, $\Sigma_i \subseteq \Sigma$ is the set of events refused (if offered) at the indicated point. Symbols P, Q, R, S, etc. are used to denote sets of refusal-traces. Refusal-traces are also referred to as *trajectories*.

Given $s \in \Sigma^*$, we use $|s|$ to denote the length of s, and for each $k \leq |s|$, $\sigma_k(s) \in \Sigma$ is used to denote the kth event in s. If $t \in \Sigma^*$ is another trace such that $|t| \leq |s|$ and for each $k \leq |t|$, $\sigma_k(t) = \sigma_k(s)$, then t is said to be a prefix of s, denoted $t \leq s$. For each $k \leq |s|$, s^k denotes the prefix of length k of s. The prefix-closure of $s \in \Sigma^*$, denoted $pr(s) \subseteq \Sigma^*$, is defined as $pr(s) := \{t \in \Sigma^* \mid t \leq s\}$. The prefix-closure map can be defined for a set of traces in a natural way.

Given $e \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$, we use $|e|$ to denote the length of e, and for each $k \leq |e|$, $\Sigma_k(e) \subseteq \Sigma$ is used to denote the kth refusal in e and $\sigma_k(e) \in \Sigma$ is used to denote the kth event in e, i.e.,

$$e = \Sigma_0(e)(\sigma_1(e), \Sigma_1(e)) \dots (\sigma_k(e), \Sigma_k(e)) \dots (\sigma_{|e|}(e), \Sigma_{|e|}(e)).$$

If $f \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$ is another refusal-trace such that $|f| \leq |e|$ and for each $k \leq |f|$, $\Sigma_k(f) = \Sigma_k(e)$ and $\sigma_k(f) = \sigma_k(e)$, then f is said to be a prefix of e, denoted $f \leq e$. For each $k \leq |e|$, e^k is used to denote the prefix of length k of e. If $f \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$ is such that $|f| = |e|$ and for each $k \leq |f|$, $\Sigma_k(f) \subseteq \Sigma_k(e)$ and $\sigma_k(f) = \sigma_k(e)$, then f is said to be dominated by e, denoted $f \sqsubseteq e$.

The prefix-closure of $e \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$, denoted $pr(e) \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, is defined as $pr(e) := \{f \in 2^\Sigma(\Sigma \times 2^\Sigma)^* \mid f \leq e\}$, and the dominance-closure of e, denoted $dom(e) \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, is defined as $dom(e) := \{f \in 2^\Sigma(\Sigma \times 2^\Sigma)^* \mid f \sqsubseteq e\}$. The prefix-closure and dominance-closure maps can be defined for a set of refusal-traces in a natural way. Given a refusal-trace $e \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$, the trace of e, denoted $tr(e) \in \Sigma^*$, is defined as $tr(e) := \sigma_1(e) \dots \sigma_{|e|}(e)$. The trace map can be extended to a set of refusal-traces in a natural way. Given a set of refusal-traces $P \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, we use $L(P) := tr(P)$ to denote its set of traces.

Symbols $\mathcal{P}, \mathcal{Q}, \mathcal{R}$, etc. are used to denote NSM's (with ϵ-moves). Let the 5-tuple

$$\mathcal{P} := (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P}, X^m_\mathcal{P})$$

represent a discrete event system modeled as an NSM, where $X_{\mathcal{P}}$ is the state set, Σ is the finite event set, $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ denotes the nondeterministic transition function[2], $x^0_{\mathcal{P}} \in X_{\mathcal{P}}$ is the initial state, and $X^m_{\mathcal{P}} \subseteq X_{\mathcal{P}}$ is the set of accepting or marked states. A triple $(x_1, \sigma, x_2) \in X_{\mathcal{P}} \times (\Sigma \cup \{\epsilon\}) \times X_{\mathcal{P}}$ is said to be a transition if $x_2 \in \delta_{\mathcal{P}}(x_1, \sigma)$. A transition (x_1, ϵ, x_2) is referred to as a *silent* or *hidden* transition. We assume that the plant cannot undergo an unbounded number of silent transitions, i.e., $\mathcal{P}$ does not contain any cycle of silent transitions. The *ϵ-closure* of $x \in X_{\mathcal{P}}$, denoted $\epsilon^*_{\mathcal{P}}(x) \subseteq X_{\mathcal{P}}$, is defined recursively as:

$$x \in \epsilon^*_{\mathcal{P}}(x); \quad [x' \in \epsilon^*_{\mathcal{P}}(x)] \Rightarrow [\delta_{\mathcal{P}}(x', \epsilon) \subseteq \epsilon^*(x)],$$

and the set of *refusal events* at $x \in X_{\mathcal{P}}$, denoted $\Re_{\mathcal{P}}(x) \subseteq \Sigma$, is defined as

$$\Re_{\mathcal{P}}(x) := \{\sigma \in \Sigma \mid \delta_{\mathcal{P}}(x', \sigma) = \emptyset, \forall x' \in \epsilon^*_{\mathcal{P}}(x)\}.$$

In other words, given $x \in X_{\mathcal{P}}$, $\epsilon^*_{\mathcal{P}}(x)$ is the set of states that can be reached from x on zero or more ϵ-moves, and $\Re_{\mathcal{P}}(x)$ is the set of events that are undefined at each state in the ϵ-closure of x.

The transition function $\delta_{\mathcal{P}} : X \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ is extended to the set of *traces* as $\delta^*_{\mathcal{P}} : X \times \Sigma^* \to 2^{X_{\mathcal{P}}}$, which is defined inductively as:

$$\forall x \in X_{\mathcal{P}} : \begin{cases} \delta^*_{\mathcal{P}}(x, \epsilon) := \epsilon^*_{\mathcal{P}}(x), \\ \forall s \in \Sigma^*, \sigma \in \Sigma : \delta^*_{\mathcal{P}}(x, s\sigma) := \epsilon^*_{\mathcal{P}}(\delta_{\mathcal{P}}(\delta^*_{\mathcal{P}}(x, s), \sigma)), \end{cases}$$

where in the last equality the transition function $\delta_{\mathcal{P}} : X \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ has been extended to $\delta_{\mathcal{P}} : 2^{X_{\mathcal{P}}} \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ in a natural way. The transition function is also extended to the set of *refusal-traces* as $\delta^T_{\mathcal{P}} : X \times (2^{\Sigma}(\Sigma \times 2^{\Sigma})^*) \to 2^{X_{\mathcal{P}}}$, which is defined inductively as:

$$\forall x \in X_{\mathcal{P}} : \begin{cases} \forall \Sigma' \subseteq \Sigma : \delta^T_{\mathcal{P}}(x, \Sigma') := \{x' \in \epsilon^*_{\mathcal{P}}(x) \mid \Sigma' \subseteq \Re_{\mathcal{P}}(x')\}, \\ \forall e \in 2^{\Sigma}(\Sigma \times 2^{\Sigma})^*, \sigma \in \Sigma, \Sigma' \subseteq \Sigma : \\ \delta^T_{\mathcal{P}}(x, e(\sigma, \Sigma')) := \{x' \in \epsilon^*_{\mathcal{P}}(\delta_{\mathcal{P}}(\delta^T_{\mathcal{P}}(x, e), \sigma)) \mid \Sigma' \subseteq \Re_{\mathcal{P}}(x')\}. \end{cases}$$

In other words, a state $x' \in X_{\mathcal{P}}$ is reached by executing a zero-length refusal-trace $\Sigma' \subseteq \Sigma$ from a state $x \in X_{\mathcal{P}}$ if x' can be reached in zero or more ϵ-moves from x, and each event in Σ' is refused at x'. A state $x' \in X_{\mathcal{P}}$ is reached by executing a refusal-trace $e(\sigma, \Sigma') \in 2^{\Sigma} \times (\Sigma \times 2^{\Sigma})^*$ from a state $x \in X_{\mathcal{P}}$ if x' can be reached by executing the event σ followed by zero or more ϵ-moves from a state reached by executing the refusal-trace e from x, and each event in Σ' is refused at state x'.

The extended transition functions are then used to obtain the language models and the trajectory models of $\mathcal{P}$ as follows:

$$L(\mathcal{P}) := \{s \in \Sigma^* \mid \delta^*_{\mathcal{P}}(x^0_{\mathcal{P}}, s) \neq \emptyset\}, \quad L^m(\mathcal{P}) := \{s \in L(\mathcal{P}) \mid \delta^*_{\mathcal{P}}(x^0_{\mathcal{P}}, s) \cap X^m_{\mathcal{P}} \neq \emptyset\},$$

$$T(\mathcal{P}) := \{e \in 2^{\Sigma}(\Sigma \times 2^{\Sigma})^* \mid \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e) \neq \emptyset\}, T^m(\mathcal{P}) := \{e \in T(\mathcal{P}) \mid \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e) \cap X^m_{\mathcal{P}} \neq \emptyset\}.$$

[2] ϵ represents both an *internal* or *unobservable* event and an *internal* or *nondeterministic* choice [7, 12].

$L(\mathcal{P}), L^m(\mathcal{P}), T(\mathcal{P}), T^m(\mathcal{P})$ are called the *generated language, recognized or marked language, generated trajectory set, recognized or marked trajectory set*, respectively, of $\mathcal{P}$. It is easily seen that $L(T^m(\mathcal{P})) = L^m(\mathcal{P})$ and $L(T(P)) = L(\mathcal{P})$. The pairs $(L^m(\mathcal{P}), L(\mathcal{P}))$ and $(T^m(\mathcal{P}), T(\mathcal{P}))$ are called the *language model* and the *trajectory model*, respectively, of $\mathcal{P}$. Two language models $(K_1^m, K_1), (K_2^m, K_2)$ are said to be equal, written $(K_1^m, K_1) = (K_2^m, K_2)$, if $K_1^m = K_2^m, K_1 = K_2$; equality of two trajectory models is defined analogously.

3 Trajectory Models

Above we defined the trajectory model of NSM's. Trajectory models are important for our eventual goal of supervisory control of nondeterministic systems since we wish to exercise control by composing systems in prioritized synchrony, and as shown in the next section trajectory models retain sufficient information about the system to allow adequate supervisory design (language model and even the failures model are not adequate as they are not detailed enough).

In order to gain further understanding into the structure of trajectory models, we next obtain a necessary and sufficient condition for a given refusal-trace set pair to be a trajectory model. This requires the definition of saturated refusal-traces. Given a refusal-trace set $P \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, we define the *saturation map* on P by $sat_P : P \to 2^\Sigma(\Sigma \times 2^\Sigma)^*$ where

$$\begin{aligned} sat_P(e) &:= \Sigma_0(\sigma_1(e), \Sigma_1) \dots (\sigma_{|e|}, \Sigma_{|e|}), \text{ where} \\ \forall k \leq |e| : \Sigma_k &:= \Sigma_k(e) \cup \{\sigma \in \Sigma \mid e^k(\sigma, \emptyset) \notin dom(pr(P))\}. \end{aligned}$$

Thus if an event is not executable in P at a certain point of its refusal-trace e, then it is added to the refusal set of e at that point to obtain $sat_P(e)$. The *saturated refusal-traces* of P, denoted $P_{sat} \subseteq P$, is defined to be the set of fixed points of $sat_P(\cdot)$. Heymann-Meyer used an axiomatic characterization for *defining* a generated trajectory set. In contrast, we proved in [16, Theorem 1] that these axioms are *necessary and sufficient* for a given refusal-trace set to be a generated trajectory set of some NSM. This we recall here:

Theorem 1 [16, Theorem 1] Given a refusal-trace set $P \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, there exists an NSM $\mathcal{P}$ with generated trajectory set P, i.e., $P = T(\mathcal{P})$, if and only if

1. $P \neq \emptyset$
2. $P = pr(P)$
3. $P = dom(P)$
4. $sat_P(P) \subseteq P$
5. $\forall e \in P : \sigma_{k+1}(e) \notin \Sigma_k(e), \quad k \leq |e| - 1$

Corollary 1 If P is a generated trajectory set, then

1. $sat_P(\cdot)$ is idempotent,
2. $P_{sat} = sat_P(P)$.

Proof: Let $e \in P$, $\hat{e} := sat_P(e)$, and $k \leq |e| = |\hat{e}|$. We need to show that if $\sigma \notin \Sigma_k(\hat{e})$, then $g := \hat{e}^k(\sigma, \emptyset) \in P$. Since $\sigma \notin \Sigma_k(\hat{e})$, $f := e^k(\sigma, \emptyset) \in P$ by definition of $sat_P(\cdot)$. Since $(sat_P(f))^k = sat_P(e^k) = \hat{e}^k$, it follows that $g \sqsubseteq sat_P(f) \in P$ by Theorem 1, Property 4. By Property 3, it follows that $g \in P$, so $sat_P(\hat{e}) = \hat{e}$. The second claim is an immediate consequence of the first. ■

Remark 1 It follows easily from Corollary 1 that Properties 3 and 4 in Theorem 1 can be replaced by the single property $P = dom(P_{sat})$. In order to see this first suppose $P = dom(P_{sat})$. Then P is dominance-closed, so we have $P = dom(P)$, i.e., T3 holds. Also, from Corollary 1, $sat_P(P) = P_{sat} \subseteq dom(P_{sat}) = P$, i.e., T4 holds. On the other hand, if T3 and T4 hold, then from Corollary 1 and T4 we have $P_{sat} = sat_P(P) \subseteq P$. Hence it follows from T3 that $dom(P_{sat}) \subseteq dom(P) = P$. The reverse containment $P \subseteq dom(P_{sat}) = dom(sat_P(P))$ follows from the definitions of the saturation map and the dominance-closure operation. ■

The following result generalizes Theorem 1 to characterize those refusal-trace set pairs that are trajectory models of NSM's. If $\Sigma_1, \ldots, \Sigma_n$ are subsets of Σ, then $min(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ denotes the set of minimal sets from among the given subsets with respect to the inclusion partial order.

Theorem 2 Given a pair of refusal-trace sets (P^m, P) with $P^m, P \subseteq 2^{\Sigma}(\Sigma \times 2^{\Sigma})^*$, it is a trajectory model if and only if

T1: $P \neq \emptyset$
T2: $P = pr(P)$
T3: $P = dom(P_{sat})$
T4: $\forall e \in P : \sigma_{k+1}(e) \notin \Sigma_k(e), \quad k \leq |e| - 1$
T5: $P^m = dom(P_{sat} \cap P^m)$

Proof: First suppose that (P^m, P) is a trajectory model, i.e., there exists an NSM $\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x^0_{\mathcal{P}}, X^m_{\mathcal{P}})$ such that $T^m(\mathcal{P}) = P^m$ and $T(\mathcal{P}) = P$. Then it follows from Theorem 1 and Remark 1 that T1 through T4 hold. In order to show that T5 also holds, we first show that $P^m = dom(P^m)$, i.e., $dom(P^m) \subseteq P^m$. Pick $e \in dom(P^m)$, then there exists $f \in P^m$ such that $e \sqsubseteq f$. Hence $\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e) \supseteq \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f)$, which implies that $\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e) \cap X^m_{\mathcal{P}} \supseteq \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f) \cap X^m_{\mathcal{P}}$, which is nonempty since $f \in P^m$. Thus $e \in P^m$, so $P^m = dom(P^m)$. Hence, $dom(P_{sat} \cap P^m) \subseteq dom(P^m) = P^m$. It remains to show that $P^m \subseteq dom(P_{sat} \cap P^m)$. We show using induction on length of refusal-traces that for each $e \in P$ and $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e)$, there exists $f \in P_{sat}$ such that $e \sqsubseteq f$ and $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f)$. If $|e| = 0$, then $e = \Sigma' \subseteq \Sigma$. If $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, \Sigma')$, then $x \in \epsilon^*_{\mathcal{P}}(x^0_{\mathcal{P}})$ and $\Sigma' \subseteq \Re_{\mathcal{P}}(x)$. Set $f := \Re_{\mathcal{P}}(x)$; then clearly, $f \in P_{sat}, e \sqsubseteq f$, and $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f)$. This proves the base step of induction. In order to prove the induction step, let $e = \bar{e}(\sigma, \Sigma') \in P$, $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e)$. Then $x \in \epsilon^*_{\mathcal{P}}(\delta_{\mathcal{P}}(\bar{x}, \sigma))$, where $\bar{x} \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, \bar{e})$ and $\Sigma' \subseteq \Re_{\mathcal{P}}(x)$. Since T2 holds, $e \in P$ implies $\bar{e} \in P$. Hence from induction hypothesis, there exists $\bar{f} \in P_{sat}$ such that $\bar{e} \sqsubseteq \bar{f}$ and $\bar{x} \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, \bar{f})$. Set $f := \bar{f}(\sigma, \Re_{\mathcal{P}}(x))$; then $f \in P_{sat}, e \sqsubseteq f$, and $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f)$. This proves the induction step. Hence it follows that given $e \in P^m$, so that there exists $x \in X^m_{\mathcal{P}}$ with $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e)$, we can select $f \in P_{sat}$ such that $e \sqsubseteq f$ and

$x \in \delta_{\mathcal{P}}^T(x_{\mathcal{P}}^0, f)$, i.e., $f \in P_{sat} \cap P^m$. Since $e \sqsubseteq f$, this implies that $e \in dom(P_{sat} \cap P^m)$ as desired.

Next assume that T1 through T5 hold. We need to show that (P^m, P) is a trajectory model, i.e., there exists an NSM $\mathcal{P}$ such that $T^m(\mathcal{P}) = P^m$ and $T(\mathcal{P}) = P$. Consider the NSM $\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x_{\mathcal{P}}^0, X_{\mathcal{P}}^m)$ (refer to Remark 2 for an explanation), where

- $X_{\mathcal{P}} := P_{sat}$,
- $x_{\mathcal{P}}^0 := \{\sigma \in \Sigma \mid \emptyset(\sigma, \emptyset) \notin P\}$,
- $X_{\mathcal{P}}^m := P_{sat} \cap P^m$,
- $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ is defined as:
  1. $\forall e \in P_{sat}, \sigma \in \Sigma$:

$$\delta_{\mathcal{P}}(e, \sigma) := \begin{cases} e(\sigma, \{\sigma' \in \Sigma \mid e(\sigma, \emptyset)(\sigma', \emptyset) \notin P\}) & \text{if } e(\sigma, \emptyset) \in P \\ \emptyset & \text{otherwise,} \end{cases}$$

  2 (a). $\forall \Sigma' \subseteq \Sigma$ such that $\Sigma' \in P_{sat}$:

$$\delta_{\mathcal{P}}(\Sigma', \epsilon) := min(\{\Sigma'' \subseteq \Sigma \mid \Sigma'' \in P_{sat}, \Sigma' \subset \Sigma''\}),$$

  2 (b). $\forall e \in 2^{\Sigma}(\Sigma \times 2^{\Sigma})^*, \sigma \in \Sigma, \Sigma' \subseteq \Sigma$ such that $e(\sigma, \Sigma') \in P_{sat}$:

$$\delta_{\mathcal{P}}(e(\sigma, \Sigma'), \epsilon) := \{e(\sigma, \Sigma'') \mid \Sigma'' \in min(\{\hat{\Sigma} \subseteq \Sigma \mid e(\sigma, \hat{\Sigma}) \in P_{sat}, \Sigma' \subset \hat{\Sigma}\})\}.$$

From [16, Lemma 1], it follows that $x_{\mathcal{P}}^0 \in P_{sat}$ and for each $e \in P_{sat}, \sigma \in \Sigma$, $\delta_{\mathcal{P}}(e, \sigma) \in P_{sat}$ whenever it is nonempty. Thus NSM $\mathcal{P}$ is well-defined. It follows from [16, Proposition 2] that $T(\mathcal{P}) = P$. It remains to show that $T^m(\mathcal{P}) = P^m$. By definition we have $T^m(\mathcal{P}) = \{e \in T(\mathcal{P}) \mid \delta_{\mathcal{P}}^T(x_{\mathcal{P}}^0, e) \cap X_{\mathcal{P}}^m \neq \emptyset\}$. Since $T(\mathcal{P}) = P, X_{\mathcal{P}}^m = P_{sat} \cap P^m$, and for each $e \in P$, $\delta_{\mathcal{P}}^T(x_{\mathcal{P}}^0, e) = \{f \in P_{sat} \mid e \sqsubseteq f\}$ [16, Corollary 1], we have

$$\begin{aligned} T^m(\mathcal{P}) &= \{e \in P \mid \{f \in P_{sat} \mid e \sqsubseteq f\} \cap (P_{sat} \cap P^m) \neq \emptyset\} \\ &= \{e \in P \mid \{f \in P_{sat} \cap P^m \mid e \sqsubseteq f\} \neq \emptyset\} \\ &= dom(P_{sat} \cap P^m) \\ &= P^m, \end{aligned}$$

where the last equality follows from T5. ■

Remark 2 In the proof of the sufficiency part of Theorem 2 the NSM $\mathcal{P}$ is constructed from a given refusal-trace set pair (P^m, P) satisfying T1-T5 as follows: The state space of $\mathcal{P}$ equals P_{sat}, the set of saturated refusal-traces of P; the marked states of $\mathcal{P}$ are those saturated refusal-traces which also belong to P^m; and the initial state of $\mathcal{P}$ is the (unique) minimal zero-length saturated refusal-trace of P. The state reached by executing a non-epsilon event $\sigma \in \Sigma$ from a state $e \in P_{sat}$ equals the minimal saturated refusal-trace of the type $e(\sigma, \Sigma')$ dominating $e(\sigma, \emptyset)$. The set of states reached by executing an epsilon

transition from a zero-length refusal-trace $\Sigma' \in P_{sat} = X_{\mathcal{P}}$ equals the set of minimal zero-length saturated refusal-traces dominating Σ'. Also, the set of states reached by executing an epsilon transition from a refusal-trace $e(\sigma, \Sigma') \in P_{sat} = X_{\mathcal{P}}$ equals the set of minimal saturated refusal-traces of the type $e(\sigma, \Sigma'')$ dominating $e(\sigma, \Sigma')$.

This NSM construction is the same as that given in [16, Algorithm 1] except that accepting states are also defined. A construction procedure somewhat similar to the above construction was first given without any proof in [6]. Our construction has the advantage that it avoids introduction of certain auxiliary states [16, Remark 3]. ■

The following result was obtained in the course of the proof of Theorem 2.

Corollary 2 Let $\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x^0_{\mathcal{P}}, X^m_{\mathcal{P}})$ be an NSM. Then for each $e \in T(\mathcal{P})$ and $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e)$, there exists $f \in (T(\mathcal{P}))_{sat}$ such that $e \sqsubseteq f$, $x \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, f)$ and $\Sigma_{|f|}(f) = \Re_{\mathcal{P}}(x)$.

The result of Theorem 2 is not a trivial generalization of Theorem 1. In fact for a language model (K^m, K), the prefix-closure of the recognized language K^m is the generated language of an appropriate state machine provided K^m is nonempty. The situation is different for a trajectory model (P^m, P). If P^m is nonempty, then its prefix-closure, $pr(P^m)$, satisfies properties T1,T2,T4. However, $pr(P^m)$ need not satisfy T3 in which case it cannot be the generated trajectory set of any NSM. (See Example 1 below.) The following result shows that a generated trajectory set can be obtained by taking *saturation closure* and provides further insight into the structure of trajectory models.

Proposition 1 Let Q be a nonempty refusal-trace set satisfying $pr(dom(Q)) = Q$ and T4. Then $R := dom(sat_Q(Q))$ is a generated trajectory set.

Proof: R is trivially nonempty and $R = dom(R)$. Since Q is prefix-closed and for each $e \in Q, k \leq |e|$, $sat_Q(e^k) = (sat_Q(e))^k$, R is prefix-closed. Also, from the definition of $sat_Q(\cdot)$, T4 holds for R since it holds for Q. It remains only to show that $sat_R(R) \subseteq R$. Since $sat_R(\cdot)$ is monotone (with respect to $\sqsubseteq$) and R is dominance-closed, it suffices to show that $sat_R(sat_Q(Q)) \subseteq R$.

Let $e \in Q$ and let $\hat{e} := sat_Q(e)$. Then by definition, $\hat{e} \in R$. Thus in order to show that $sat_R(\hat{e}) \in R$, it suffices to show $sat_R(\hat{e}) = \hat{e}$. We need to show that if there exist k, σ such that $\sigma \notin \Sigma_k(\hat{e})$, then $\hat{e}^k(\sigma, \emptyset) \in R$. Since $\sigma \notin \Sigma_k(\hat{e})$, it follows that $f := e^k(\sigma, \emptyset) \in Q$. Since the map $sat_Q(\cdot)$ commutes with the operation of taking the length-k prefix, we have

$$\hat{e}^k(\sigma, \emptyset) = (sat_Q(e))^k(\sigma, \emptyset) = sat_Q(e^k)(\sigma, \emptyset) \sqsubseteq sat_Q(f) \in R.$$

This shows that $sat_R(\hat{e}) = \hat{e}$, completing the proof. ■

Corollary 3 Let P^m be a nonempty recognized trajectory set. Then $dom(sat_{pr(P^m)}(pr(P^m))$ is a generated trajectory set.

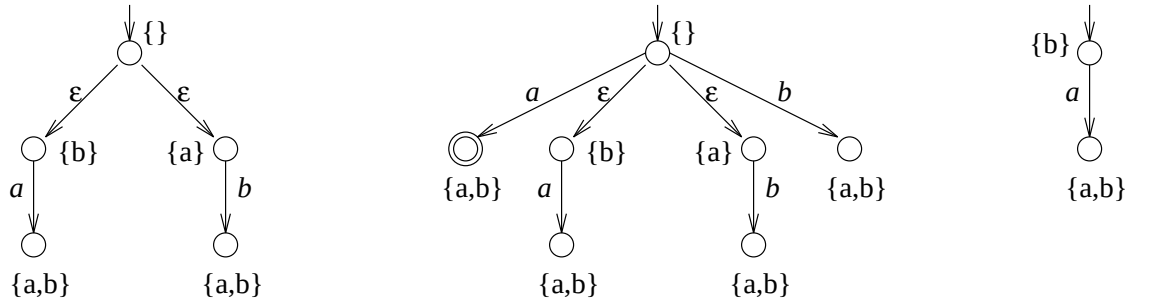


Figure 1: Diagram illustrating Example 1

Example 1 Consider the NSM $\mathcal{P}$ with $\Sigma = \{a, b\}$ and unspecified marking shown in Figure 1(a); each state is labeled with the set of events that are refused at that state, i.e., the set of events that are not executable following zero or more silent transitions. Thus, $\mathcal{P}$ is obtained from the nondeterministic choice between two deterministic subsystems–one that executes a and deadlocks, the other that executes b and deadlocks. Then the generated trajectory set of $\mathcal{P}$ is given by $P := T(\mathcal{P}) = dom(pr(\{e_1, e_2\}))$, where $e_1 := \{b\}(a, \{a, b\}),\ e_2 := \{a\}(b, \{a, b\})$. The saturated refusal-traces of P are given by

$$P_{sat} = \{\emptyset, \{a\}, \{b\}, e_1, e_2, e_3, e_4\},$$

where $e_3 := \emptyset(a, \{a, b\}),\ e_4 := \emptyset(b, \{a, b\})$. Let $P^m := dom(\{e_3\})$. Then $dom(P^m \cap P_{sat}) = dom(\{e_3\}) = P^m$, so that the refusal-trace set pair (P^m, P) satisfies T1-T5. Hence it follows from Theorem 2 that there exists a NSM $\mathcal{Q}$ such that $T(\mathcal{Q}) = P,\ T^m(\mathcal{Q}) = P^m$. One choice for $\mathcal{Q}$ is the canonical NSM described in the proof of Theorem 2, which is shown in Figure 1(b). However, there is no marking of the states of $\mathcal{P}$ for which $T^m(\mathcal{P}) = P^m$. Thus a specification of the marking information results in "refinement" of the associated state machine.

Furthermore, if we consider $P' := pr(P^m) = pr(dom(\{e_3\})) = \{\emptyset\} \cup dom(\{e_3\})$, then we have $sat_{P'}(P') = pr(e_1)$. Clearly, $dom(sat_{P'}(P')) = dom(pr(e_1)) \neq P'$, i.e., T3 does not hold for P'; consequently, it cannot be the generated trajectory set of any NSM. However, it follows from Corollary 3 that $dom(sat_{P'}(P')) = dom(pr(e_1))$ is a generated trajectory set. The NSM $\mathcal{R}$ shown in Figure 1(c) generates this trajectory set. ■

Next we identify the trajectory models of deterministic state machines. This is used later for designing supervisors which have deterministic state machine representation.

Definition 1 A trajectory model (P^m, P) is said to be *deterministic* if there exists a deterministic state machine $\mathcal{P} := (X_{\mathcal{P}}, \Sigma, \delta_{\mathcal{P}}, x^0_{\mathcal{P}}, X^m_{\mathcal{P}})$ such that $T^m(\mathcal{P}) = P^m$ and $T(\mathcal{P}) = P$.

An equivalent definition of a deterministic generated trajectory set was first given in [6, definition 12.4] (refer to Remark 3 below). Note that given a trajectory model, the trace map

can be used to obtain the associated language model. Conversely, given a language model (K^m, K), the trajectory map $trj_K : K \to 2^\Sigma(\Sigma \times 2^\Sigma)^*$ can be used to obtain the associated deterministic trajectory model:

$$\begin{aligned} trj_K(s) &:= \Sigma_0(s)(\sigma_1(s), \Sigma_1(s)) \ldots (\sigma_{|s|}(s), \Sigma_{|s|}(s)), \text{ where} \\ \Sigma_k(s) &:= \{\sigma \in \Sigma \mid s^k\sigma \notin K\}, \forall k \leq |s|. \end{aligned}$$

Thus the kth refusal-set in the refusal-trace $trj_K(s)$ is the set of events that are unexecutable in K after the prefix of length k of s. Let $det(K) := dom(trj_K(K))$ and $det^m(K^m, K) := dom(trj_K(K^m))$.

Proposition 2 Given a language model (K^m, K), $(det^m(K^m, K), det(K))$ is the unique deterministic trajectory model with language model (K^m, K).

Proof: From a standard result, there exists a deterministic state machine $\mathcal{P}$ such that $L^m(\mathcal{P}) = K^m$ and $L(\mathcal{P}) = K$. Let (P^m, P) be the trajectory model of $\mathcal{P}$. By [16, Proposition 3], $det(K)$ is the unique deterministic generated trajectory model with generated language K, so $P = det(K)$. It is clear from the definition of trj_K that $P_{sat} = trj_K(K)$. By T5, it follows that $P^m = dom(trj_K(K) \cap P^m)$. Thus, $K^m = L(P^m) = L(trj_K(K) \cap P^m)$, which implies that $trj_K(K) \cap P^m = trj_K(K^m)$, so $P^m = det^m(K^m, K)$. ■

Remark 3 It follows from Proposition 2 and [6, Proposition 12.5] that the definition of a deterministic generated trajectory set given in [6, Definition 12.4] and the definition given above are in fact equivalent. ■

4 Prioritized Synchronization and Augmentation

In this section we formally introduce prioritized synchronous composition (PSC) which is used for interconnecting various systems including plant and supervisor. We show that trajectory model retains sufficient information to infer the adequate behavior of the interconnected system for the behavior of the component systems. For PSC, each system is assigned a priority set of events. When systems are interconnected via PSC, an event can occur in the composite system only if it can occur in each subsystem which has priority over it. In this way, a subsystem can prevent the occurrence of certain events, thereby implementing a type of supervisory control. The following definition of PSC of two NSM's trivially extends the one given by us in [16, Definition 9] by incorporating the notion of markings. The initial definition given by Heymann [4] was limited to NSM's without epsilon-moves.

Definition 2 Let $\mathcal{P} := (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P}, X^m_\mathcal{P})$, $\mathcal{Q} := (X_\mathcal{Q}, \Sigma, \delta_\mathcal{Q}, x^0_\mathcal{Q}, X^m_\mathcal{Q})$ be two NSM's having priority sets $A, B \subseteq \Sigma$ respectively. The PSC of $\mathcal{P}$ and $\mathcal{Q}$ is another NSM which is denoted by

$$\mathcal{P} \,{}_A\|_B\, \mathcal{Q} := \mathcal{R} := (X_\mathcal{R}, \Sigma, \delta_\mathcal{R}, x^0_\mathcal{R}, X^m_\mathcal{R}),$$

where $X_{\mathcal{R}} = X_{\mathcal{P}} \times X_{\mathcal{Q}}, x^0_{\mathcal{R}} = (x^0_{\mathcal{P}}, x^0_{\mathcal{Q}}), X^m_{\mathcal{R}} = X^m_{\mathcal{P}} \times X^m_{\mathcal{Q}}$, and the state transition function $\delta_{\mathcal{R}} : X_{\mathcal{R}} \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{R}}}$ is defined as:

$\forall x_r = (x_p, x_q) \in X_{\mathcal{R}}$:

$$\forall \sigma \in \Sigma : \delta_{\mathcal{R}}(x_r, \sigma) := \begin{cases} \delta_{\mathcal{P}}(x_p, \sigma) \times \delta_{\mathcal{Q}}(x_q, \sigma) & \text{if } \delta_{\mathcal{P}}(x_p, \sigma), \delta_{\mathcal{Q}}(x_q, \sigma) \neq \emptyset \\ \delta_{\mathcal{P}}(x_p, \sigma) \times \{x_q\} & \text{if } \delta_{\mathcal{P}}(x_p, \sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{Q}}(x_q), \sigma \notin B \\ \{x_p\} \times \delta_{\mathcal{Q}}(x_q, \sigma) & \text{if } \delta_{\mathcal{Q}}(x_q, \sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{P}}(x_p), \sigma \notin A \\ \emptyset & \text{otherwise,} \end{cases}$$

$$\delta_{\mathcal{R}}(x_r, \epsilon) := [\delta_{\mathcal{P}}(x_p, \epsilon) \cup \{x_p\}] \times [\delta_{\mathcal{Q}}(x_q, \epsilon) \cup \{x_q\}] - \{(x_p, x_q)\}.$$

Thus an event is executed synchronously whenever both systems can participate; however, it can occur asynchronously whenever one of the systems can participate and the second system refuses it but has no priority over it.

The above definition gives the NSM resulting by composing the given NSM's. However, when only the trajectory models of the two NSM's are available, in order to obtain the trajectory model of the composed system we need to define the composition of the trajectory models, and show that this definition is consistent with the definition of composition for NSM's. This is what we do next. For notational convenience, given $\Sigma', \Sigma_1, \Sigma_2, \Sigma'' \subseteq \Sigma$, we define

$$\Sigma' {}_{\Sigma_1}\bigotimes_{\Sigma_2} \Sigma'' := (\Sigma' \cap \Sigma'') \cup (\Sigma' \cap \Sigma_1) \cup (\Sigma'' \cap \Sigma_2).$$

Given *generated* trajectory sets P, Q, the PSC of a pair of trajectories $e_p \in P,\ e_q \in Q$ was first given in [6, Definition 13.1], which was made precise in [16, Definition 10] as follows:

Definition 3 Let P, Q be generated trajectory sets with $e_p \in P,\ e_q \in Q$. Then the PSC of e_p and e_q (with respect to P and Q), denoted $e_p \ {}_A\|_B \ e_q$, is defined inductively on $|e_p| + |e_q|$ as follows:

$\forall \Sigma_p, \Sigma_q \subseteq \Sigma$ such that $\Sigma_p \in P, \Sigma_q \in Q$:

$$\Sigma_p \ {}_A\|_B \ \Sigma_q := \{\Sigma' \subseteq \Sigma_p \, {}_A\bigotimes_B \, \Sigma_q\},$$

$\forall e_p \in P, e_q \in Q$ with $|e_p| + |e_q| \geq 1$:
(Let $\sigma_p := \sigma_{|e_p|}(e_p), \sigma_q := \sigma_{|e_q|}(e_q), \Sigma_p := \Sigma_{|e_p|}(e_p), \Sigma_q := \Sigma_{|e_q|}(e_q)$)

$$e_p \ {}_A\|_B \ e_q := T_1 \cup T_2 \cup T_3,$$

where

$$T_1 := \begin{cases} \{e(\sigma_p, \Sigma') \mid e \in e_p^{|e_p|-1} \ {}_A\|_B \ e_q; \Sigma' \subseteq \Sigma_p \, {}_A\bigotimes_B \, \Sigma_q\} & \text{if } |e_p| \geq 1, \sigma_p \notin B \text{ and } e_q(\sigma_p, \emptyset) \notin Q \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_2 := \begin{cases} \{e(\sigma_q, \Sigma') \mid e \in e_p;_A \|_B\ e_q^{|e_q|-1}; \Sigma' \subseteq \Sigma_p\ {}_A\bigotimes_B \Sigma_q\} & \text{if } |e_q| \geq 1, \sigma_q \notin A \text{ and } e_p(\sigma_q, \emptyset) \notin P \\ \emptyset & \text{otherwise} \end{cases}$$

$$T_3 := \begin{cases} \{e(\sigma, \Sigma') \mid e \in e_p^{|e_p|-1}\ {}_A\|_B\ e_q^{|e_q|-1}; \Sigma' \subseteq \Sigma_p\ {}_A\bigotimes_B \Sigma_q\} & \text{if } |e_p|, |e_q| \geq 1 \text{ and } \sigma_p = \sigma_q := \sigma \\ \emptyset & \text{otherwise} \end{cases}$$

It should be noted that $e_p\ {}_A\|_B\ e_q$ is a set of refusal-traces that depends on the generated trajectory sets P, Q as well as on the particular trajectories e_p, e_q. The dependence on P, Q is not explicitly indicated in the notation.

The PSC of two zero-length refusal-traces $\Sigma_p \in P$ and $\Sigma_q \in Q$ is obtained by computing $\Sigma_p\ {}_A\bigotimes_B \Sigma_q = (\Sigma_p \cap \Sigma_q) \cup (\Sigma_p \cap A) \cup (\Sigma_q \cap B)$, i.e., an event is refused in the composed system if either it is refused in both the systems, or it is refused in a system which has priority over that event. Next the PSC of two refusal-traces $e_p \in P, e_q \in Q$ with at least one of them of length more than zero (so that at least one of them has the form: $e_p = e_p^{|e_p|-1}(\sigma_p, \Sigma_p), e_q = e_q^{|e_q|-1}(\sigma_q, \Sigma_q)$), is obtained by considering these three possible cases: (i) e_p is of length more than zero and a refusal-trace belonging to $e_p^{|e_p|-1}\ {}_A\|_B\ e_q$ has already been executed in the composed system, and at this point, the occurrence of the last event of e_p cannot be blocked by Q (indicated by $\sigma_p \notin B$), and Q cannot participate in the occurrence of this event (indicated by $e_q(\sigma_p, \emptyset) \notin Q$); (ii) e_p is of length more than zero and a refusal-trace belonging to $e_p\ {}_A\|_B\ e_q^{|e_q|-1}$ has already been executed in the composed system, and at this point, P can neither block the occurrence of the last event of e_q, nor it can participate in its occurrence; (iii) Both e_p and e_q are of length more than zero and their final events are same (indicated by $\sigma_p = \sigma_q := \sigma$); a refusal-trace belonging to $e_p^{|e_p|-1}\ {}_A\|_B\ e_q^{|e_q|-1}$ has already been executed in the composed system, and at this point, σ is executable in both P and Q.

The definition of PSC of refusal-traces is extended to obtain the definition of the PSC of trajectory models:

Definition 4 Let (P^m, P), (Q^m, Q) be trajectory models with priority sets $A, B \subseteq \Sigma$ respectively. The PSC of (P^m, P) and (Q^m, Q), denoted $(P^m, P)\ {}_A\|_B\ (Q^m, Q)$, is the pair of refusal-trace sets (R^m, R), where

$$R^m := \bigcup_{e_p \in P^m, e_q \in Q^m} e_p\ {}_A\|_B\ e_q, \quad R := \bigcup_{e_p \in P, e_q \in Q} e_p\ {}_A\|_B\ e_q,$$

where $e_p\ {}_A\|_B\ e_q$ is with respect to P, Q in the definitions of both R^m and R.

We will use the notation $(P^m\ {}_A\|_B\ Q^m, P\ {}_A\|_B\ Q)$ for $(P^m, P)\ {}_A\|_B\ (Q^m, Q)$. However, it must be kept in mind that $P^m\ {}_A\|_B\ Q^m$ implicitly depends on P and Q. The following theorem proves that the trajectory model retains sufficient system detail to support prioritized

synchronous composition. This result justifies the need for using trajectory models for representing nondeterministic systems that are interacting via prioritized synchronization such as plant and supervisor in the setting of supervisory control. (Heymann [4] showed that a corresponding result does not hold for less detailed models such as language or failures model.)

Theorem 3 Let $\mathcal{P}, \mathcal{Q}$ be NSM's and $A, B \subseteq \Sigma$. $T^m(\mathcal{P})\ {}_A\|_B\ T^m(\mathcal{Q}) = T^m(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})$ and $T(\mathcal{P})\ {}_A\|_B\ T(\mathcal{Q}) = T(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})$.

Proof: The fact that $T(\mathcal{P})\ {}_A\|_B\ T(\mathcal{Q}) = T(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})$ is proved in [16, Theorem 2]. For notational convenience, let $\mathcal{R} := \mathcal{P}\ {}_A\|_B\ \mathcal{Q}$; we first prove that $T^m(\mathcal{R}) \subseteq T^m(\mathcal{P})\ {}_A\|_B\ T^m(\mathcal{Q})$. Pick $e \in T^m(\mathcal{R})$. Then $e \in T(\mathcal{R})$, and there exists $x_r = (x_p, x_q) \in \delta^T_{\mathcal{R}}(x^0_{\mathcal{R}}, e) \cap X^m_{\mathcal{R}} = \delta^T_{\mathcal{R}}(x^0_{\mathcal{R}}, e) \cap (X^m_{\mathcal{P}} \times X^m_{\mathcal{Q}})$. It follows from [16, Corollary 5] that given e and x_r, there exists $e_p \in T(\mathcal{P})$ and $e_q \in T(\mathcal{Q})$ such that

$$e \in e_p\ {}_A\|_B\ e_q \text{ and } x_r \in \delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \times \delta^T_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q). \tag{1}$$

It follows that $\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \cap X^m_{\mathcal{P}} \neq \emptyset$ and $\delta^T_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q) \cap X^m_{\mathcal{Q}} \neq \emptyset$. This implies that $e_p \in T^m(\mathcal{P})$ and $e_q \in T^m(\mathcal{Q})$, proving that $e \in T^m(\mathcal{P})\ {}_A\|_B\ T^m(\mathcal{Q})$.

Next we prove that $T^m(\mathcal{P})\ {}_A\|_B\ T^m(\mathcal{Q}) \subseteq T^m(\mathcal{R})$. Pick $e \in T^m(\mathcal{P})\ {}_A\|_B\ T^m(\mathcal{Q})$. Then there exist $e_p \in T^m(\mathcal{P})$ and $e_q \in T^m(\mathcal{Q})$ such that $e \in e_p\ {}_A\|_B\ e_q$. Since $e_p \in T(\mathcal{P})$ and $e_q \in T(\mathcal{Q})$, it follows from [16, Corollary 5] that $\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \times \delta^T_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q) \subseteq \delta_{\mathcal{R}}(x^0_{\mathcal{R}}, e)$. This implies that

$$[\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \cap X^m_{\mathcal{P}}] \times [\delta^T_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q) \cap X^m_{\mathcal{Q}}] \subseteq \delta^T_{\mathcal{R}}(x^0_{\mathcal{R}}, e) \cap X^m_{\mathcal{R}}. \tag{2}$$

Since $e_p \in T^m(\mathcal{P})$ and $e_q \in T^m(\mathcal{Q})$, we have $\delta^T_{\mathcal{P}}(x^0_{\mathcal{P}}, e_p) \cap X^m_{\mathcal{P}} \neq \emptyset$ and $\delta^T_{\mathcal{Q}}(x^0_{\mathcal{Q}}, e_q) \cap X^m_{\mathcal{Q}} \neq \emptyset$. Thus, Equation 2 implies that $\delta^T_{\mathcal{R}}(x^0_{\mathcal{R}}, e) \cap X^m_{\mathcal{R}} \neq \emptyset$, so $e \in T^m(\mathcal{R})$. ■

Next we prove the *associativity* property of PSC. This result is quite useful in the setting of supervisory control since a plant or a supervisor may be composed of several "sub-plants" or "sub-supervisors" that are interacting via prioritized synchronization, and their behavior should not depend on the order in which they are composed. First note from Definition 2 it is immediate that for NSM's $\mathcal{P}, \mathcal{Q}$ with priority sets $A, B \subseteq \Sigma$ respectively, the following holds for each state $x_r = (x_p, x_q) \in X_{\mathcal{P}} \times X_{\mathcal{Q}}$ of the composed system $\mathcal{R} := \mathcal{P}\ {}_A\|_B\ \mathcal{Q}$:

$$\Re_{\mathcal{R}}(x_r) = \Re_{\mathcal{P}}(x_p)\ {}_A\otimes_B\ \Re_{\mathcal{Q}}(x_q) = [\Re_{\mathcal{P}}(x_p) \cap \Re_{\mathcal{Q}}(x_q)] \cup [\Re_{\mathcal{P}}(x_p) \cap A] \cup [\Re_{\mathcal{Q}}(x_q) \cap B].$$

Theorem 4 Let $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ be NSM's and $A, B, C \subseteq \Sigma$. Then

$$(\mathcal{P}\ {}_A\|_B\ \mathcal{Q})\ {}_{A\cup B}\|_C\ \mathcal{R} = \mathcal{P}\ {}_A\|_{B\cup C}\ (\mathcal{Q}\ {}_B\|_C\ \mathcal{R}).$$

Proof: Let $\mathcal{S}_1 := (\mathcal{P}\ {}_A\|_B\ \mathcal{Q})\ {}_{A\cup B}\|_C\ \mathcal{R}$, and $\mathcal{S}_2 := \mathcal{P}\ {}_A\|_{B\cup C}\ (\mathcal{Q}\ {}_B\|_C\ \mathcal{R})$. Then it follows from Definition 2 that $X_{\mathcal{S}_1} = X_{\mathcal{S}_2} = X_{\mathcal{P}} \times X_{\mathcal{Q}} \times X_{\mathcal{R}} := X_{\mathcal{S}}$, $x^0_{\mathcal{S}_1} = x^0_{\mathcal{S}_2} = (x^0_{\mathcal{P}}, x^0_{\mathcal{Q}}, x^0_{\mathcal{R}})$, $X^m_{\mathcal{S}_1} = X^m_{\mathcal{S}_2} = X^m_{\mathcal{P}} \times X^m_{\mathcal{Q}} \times X^m_{\mathcal{R}}$, and for each $x_s = (x_p, x_q, x_r) \in X_{\mathcal{S}}$:

$$\delta_{\mathcal{S}_1}(x_s, \epsilon) = \delta_{\mathcal{S}_2}(x_s, \epsilon) = [\delta_{\mathcal{P}}(x_p, \epsilon) \cup \{x_p\}] \times [\delta_{\mathcal{Q}}(x_q, \epsilon) \cup \{x_q\}] \times [\delta_{\mathcal{R}}(x_r, \epsilon) \cup \{x_q\}] - \{x_s\}.$$

It remains to show that for each $x_s = (x_p, x_q, x_r) \in X_{\mathcal{S}}$ and $\sigma \in \Sigma$:

$$\delta_{\mathcal{S}_1}(x_s, \sigma) = \delta_{\mathcal{S}_2}(x_s, \sigma). \tag{3}$$

It follows from Definition 2 that

$$\delta_{\mathcal{S}_1}(x_s,\sigma) = \begin{cases} \delta_{\mathcal{P}}(x_p,\sigma) \times \delta_{\mathcal{Q}}(x_q,\sigma) \times \delta_{\mathcal{R}}(x_r,\sigma) & \text{if } \delta_{\mathcal{P}}(x_p,\sigma), \delta_{\mathcal{Q}}(x_q,\sigma), \delta_{\mathcal{R}}(x_r,\sigma) \neq \emptyset \\ \delta_{\mathcal{P}}(x_p,\sigma) \times \{x_q\} \times \delta_{\mathcal{R}}(x_r,\sigma) & \text{if } \delta_{\mathcal{P}}(x_p,\sigma), \delta_{\mathcal{R}}(x_r,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{Q}}(x_q), \sigma \notin B \\ \{x_p\} \times \delta_{\mathcal{Q}}(x_q,\sigma) \times \delta_{\mathcal{R}}(x_r,\sigma) & \text{if } \delta_{\mathcal{Q}}(x_q,\sigma), \delta_{\mathcal{R}}(x_r,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{P}}(x_p), \sigma \notin A \\ \delta_{\mathcal{P}}(x_p,\sigma) \times \delta_{\mathcal{Q}}(x_q,\sigma) \times \{x_r\} & \text{if } \delta_{\mathcal{P}}(x_p,\sigma), \delta_{\mathcal{Q}}(x_q,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{R}}(x_r), \sigma \notin C \\ \delta_{\mathcal{P}}(x_p,\sigma) \times \{x_q\} \times \{x_r\} & \text{if } \delta_{\mathcal{P}}(x_p,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{Q}}(x_q) \cap \Re_{\mathcal{R}}(x_r), \sigma \notin B \cup C \\ \{x_p\} \times \delta_{\mathcal{Q}}(x_q,\sigma) \times \{x_r\} & \text{if } \delta_{\mathcal{Q}}(x_q,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{P}}(x_p) \cap \Re_{\mathcal{R}}(x_r), \sigma \notin A \cup C \\ \{x_p\} \times \{x_q\} \times \delta_{\mathcal{R}}(x_r,\sigma) & \text{if } \delta_{\mathcal{R}}(x_r,\sigma) \neq \emptyset, \sigma \in \Re_{\mathcal{P}}(x_p) \cap \Re_{\mathcal{Q}}(x_q), \sigma \notin A \cup B \\ \emptyset & \text{otherwise} \end{cases}$$

An expression for $\delta_{\mathcal{S}_2}(x_s, \sigma)$ can be analogously obtained. Note that in the fifth clause of the expression for $\delta_{\mathcal{S}_1}(x_s, \sigma)$, the condition $\sigma \in \Re_{\mathcal{Q}}(x_q) \cap \Re_{\mathcal{R}}(x_r)$ and $\sigma \notin B \cup C$ is equivalent to $\sigma \in \Re_{\mathcal{Q}}(x_q) \,{}_B\!\otimes_C \Re_{\mathcal{R}}(x_r)$ and $\sigma \notin B \cup C$. Similarly, in the sixth clause of the expression for $\delta_{\mathcal{S}_1}(x_s, \sigma)$, the condition $\sigma \in \Re_{\mathcal{P}}(x_p) \cap \Re_{\mathcal{R}}(x_r)$ and $\sigma \notin A \cup C$ is equivalent to $\sigma \in \Re_{\mathcal{P}}(x_p) \,{}_A\!\otimes_C \Re_{\mathcal{R}}(x_r)$ and $\sigma \notin A \cup C$. If similar simplifications in the clauses of the expression for $\delta_{\mathcal{S}_2}(x_s, \sigma)$ are performed, then Equation 3 is easily proved. ■

Appendix A lists some of the corollaries of Theorems 3 and 4. We conclude this section by extending the notion of *augmentation* first introduced in [16, Subsection 5.2]. The notion of augmentation is quite useful as under certain mild conditions that are met in the setting of supervisory control, the PSC of systems can be reduced to SSC (strict synchronous composition) of appropriately augmented systems. The underlying idea of augmentation is quite simple: Since an event that belongs to the priority set of a single system can occur asynchronously, if we augment the other system by adding self-loops on such events, then the operation of PSC reduces to that of SSC of augmented systems over those events that belong to the priority set of at least one system. Formally, augmentation of an NSM with a given event set $D \subseteq \Sigma$ is its PSC with the NSM $\mathcal{D} := (\{x^0_{\mathcal{D}}\}, \Sigma, \delta_{\mathcal{D}}, x^0_{\mathcal{D}}, \{x^0_{\mathcal{D}}\})$, the transition function of which is defined as:

$$\forall \sigma \in \Sigma \cup \{\epsilon\}, \quad \delta_{\mathcal{D}}(x^0_{\mathcal{D}}, \sigma) := \begin{cases} \{x^0_{\mathcal{D}}\} & \text{if } \sigma \in D \\ \emptyset & \text{otherwise.} \end{cases}$$

In other words, $\mathcal{D}$ is a deterministic state machine consisting of a single marked state having self-loops on events in $D \subseteq \Sigma$. It is clear that $L^m(\mathcal{D}) = L(\mathcal{D}) = D^*$ and $T^m(\mathcal{D}) = T(\mathcal{D}) = det(D^*)$.

Definition 5 Given an NSM $\mathcal{P} := (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P}, X^m_\mathcal{P})$ and $D \subseteq \Sigma$, the *augmented NSM with respect to* D, denoted $\mathcal{P}^D$, is defined to be $\mathcal{P}^D := \mathcal{P} \,{}_\emptyset\|_\emptyset\, \mathcal{D}$. Given a trajectory model (P^m, P), *the augmented trajectory model with respect to* D, denoted $(P^m, P)^D$, is defined to be $(P^m, P)^D := (P^m, P) \,{}_\emptyset\|_\emptyset\, (det(D^*), det(D^*))$.

It follows from the above definition that the augmented NSM $\mathcal{P}^D$ is obtained by adding self-loops at each state of $\mathcal{P}$ on those events in D that are refused at that state, i.e., $\mathcal{P}^D := (X_\mathcal{P}, \Sigma, \delta_{\mathcal{P}^D}, x^0_\mathcal{P}, X^m_\mathcal{P})$, where the transition function is defined as:

$$\forall x \in X_\mathcal{P}, \sigma \in \Sigma \cup \{\epsilon\} : \delta_{\mathcal{P}^D}(x,\sigma) := \begin{cases} \delta_\mathcal{P}(x,\sigma) & \text{if } \delta_\mathcal{P}(x,\sigma) \neq \emptyset \\ \{x\} & \text{if } \sigma \in D \cap \Re_\mathcal{P}(x) \\ \emptyset & \text{otherwise} \end{cases}$$

Using $((P^m)^D, P^D)$ to denote $(P^m, P)^D$, we have $(P^m)^D = P^m \,{}_\emptyset\|_\emptyset\, det(D^*)$ and $P^D = P \,{}_\emptyset\|_\emptyset\, det(D^*)$. As is always the case with the PSC of *recognized* trajectory sets, the expression for $(P^m)^D$ implicitly depends on the corresponding *generated* trajectory sets P and $det(D^*)$. Clearly the trajectory model $(T^m(\mathcal{P}^D), T(\mathcal{P}^D))$ of the augmented NSM $\mathcal{P}^D$ is equal to $((T^m(\mathcal{P}))^D, (T(\mathcal{P}))^D)$. Also, since $det(D^*)$ can always execute every event in D and can never execute any event in $\Sigma - D$, it follows that given any trajectory model (P^m, P), any $A \subseteq \Sigma - D$, and any $B \subseteq D$

$$\begin{aligned} (P^m)^D &:= P^m \,{}_\emptyset\|_\emptyset\, det(D^*) = P^m \,{}_A\|_B\, det(D^*), \\ P^D &:= P \,{}_\emptyset\|_\emptyset\, det(D^*) = P \,{}_A\|_B\, det(D^*). \end{aligned}$$

The following result shows that the PSC of two systems is equivalent to the SSC (over the union of the two priority sets) of the associated augmented systems.

Proposition 3 Let $(P^m, P), (Q^m, Q)$ be trajectory models, and $A, B \subseteq \Sigma$. Then

1. $P^m \,{}_A\|_B\, Q^m = (P^m)^{B-A} \,{}_{A\cup B}\|_B\, Q^m = (P^m)^{B-A} \,{}_{A\cup B}\|_{A\cup B}\, (Q^m)^{A-B}$,
2. $P \,{}_A\|_B\, Q = P^{B-A} \,{}_{A\cup B}\|_B\, Q = P^{B-A} \,{}_{A\cup B}\|_{A\cup B}\, Q^{A-B}$.

Proof: We only prove the first part; the second part can be proved analogously. Again, we only prove the first equality as the second equality follows from symmetry and a second application of the first equality. From the definition of augmentation and associativity of PSC (Corollary 8), we have

$$\begin{aligned} (P^m)^{B-A} \,{}_{A\cup B}\|_B\, Q^m &= (P^m \,{}_A\|_{B-A}\, det((B-A)^*)) \,{}_{A\cup B}\|_B\, Q^m \\ &= det((B-A)^*) \,{}_{B-A}\|_{A\cup B}\, (P^m \,{}_A\|_B\, Q^m) \\ &= P^m \,{}_A\|_B\, Q^m, \end{aligned}$$

where the last equality follows from the two facts: (i) The priority set of $P^m \,{}_A\|_B\, Q^m$ is $A \cup B$, and $B - A \subseteq A \cup B$, so $det((B-A)^*)$ cannot execute an event that $P^m \,{}_A\|_B\, Q^m$ does not execute. (ii) $det((B-A)^*)$ can always execute each event in its priority set, so that it cannot block any event in $P^m \,{}_A\|_B\, Q^m$. ∎

Note that when $A \cup B = \Sigma$ as in the setting of supervisory control, then PSC can be transformed into SSC of augmented systems using the result of the above proposition.

5 Supervisory Control Using PSC

Supervisory control theory for nondeterministic systems in the setting of trajectory models and PSC proposed by Heymann [4] and Heymann-Meyer [6] was rigorously developed in our previous work [16]. In this work, all refusal-traces of the plant were considered marked and the desired behavior was specified by a prefix-closed language; thus the issue of blocking was not investigated. In this section, we generalize the results in [16] to include non-closed specifications and arbitrary marking of the plant refusal-traces.

Since trajectory models contain sufficient detail to support the operation of prioritized synchronization, we use trajectory models, rather than NSM's, to represent a discrete event system. Unless otherwise specified, the trajectory model of the plant and that of the supervisor are denoted by (P^m, P) and (S^m, S), and the priority set of the plant and that of the supervisor are denoted by A and B respectively. The controlled (or closed-loop) system is $(P^m, P)\,{}_A\|_B\,(S^m, S)$. In certain situations, a plant may consist of several "sub-plants" operating in prioritized synchrony. In such a case, if (P_i^m, P_i) is the trajectory model of the ith sub-plant, and $A_i \subseteq \Sigma$ is its priority set, then the trajectory model of the composed plant is given by $(P^m, P) := (\|_{A_i} P_i^m, \|_{A_i} P_i)$, where the notation $(\|_{A_i} P_i^m, \|_{A_i} P_i)$ is used to denote the PSC of sub-plants $\{(P_i^m, P_i)\}$. (Since PSC is associative (Corollary 8), $(\|_{A_i} P_i^m, \|_{A_i} P_i)$ is well defined, and it follows from Corollary 5 that it is a trajectory model.) The priority set of the composed plant is given by $A := \cup_i A_i$.

In the setting of supervisory control, $A = \Sigma_u \cup \Sigma_c$, where $\Sigma_u, \Sigma_c \subseteq \Sigma$ denote the sets of *uncontrollable* and *controllable* events respectively [14, 15]; and $B = \Sigma_c \cup \Sigma_d$, where $\Sigma_d \subseteq \Sigma$ denotes the set of so-called *driven* [4] or *forcible* [3, 2] or *command* events [1]. By definition Σ_u, Σ_c and Σ_d are pairwise disjoint and exhaust the entire event set, i.e., $A \cup B = \Sigma$. Since controllable events belong to the common priority set, they cannot occur solely in the plant or in the supervisor. This is consistent with the definition of controllable events in Ramadge-Wonham theory [14] where although they originate in the plant, their execution requires enablement by the supervisor. Only the plant has priority over the uncontrollable events, so they can occur whenever the plant can participate. Again this is consistent with the definition of uncontrollable events in Ramadge-Wonham theory where uncontrollable events are always enabled by the supervisor. Driven events are "dual" to uncontrollable events. Only the supervisor has priority over them, so they can occur whenever the supervisor is able to execute them, regardless of whether the plant can participate.

We begin by defining the notions of non-markingness and non-blockingness which are desirable properties of supervisors. Informally, a supervisor is said to be non-marking if it does not "affect" the marking status of a certain refusal-trace, i.e., if the execution of a certain refusal-trace of a controlled plant leads to a marked state of the plant, then regardless of what state is reached in the supervisor that refusal-trace should be a marked refusal-trace of the controlled plant. On the other hand, a supervisor is said to be non-blocking if it is always possible to reach a marked state of the controlled plant from any of its other states, i.e., the controlled system does not get "blocked" in one of its states that is not marked.

Definition 6 Given a plant (P^m, P) with priority set $A \subseteq \Sigma$, a supervisor, with trajectory model (S^m, S) and priority set $B \subseteq \Sigma$, is said to be *non-marking* if $S^m = S$; it is said to be *language model non-blocking* if $pr(L(P^m{}_A\|_B\, S^m)) = L(P{}_A\|_B\, S)$; and it is said to be *trajectory model non-blocking* if $pr(P^m{}_A\|_B\, S^m) = P{}_A\|_B\, S$.

Note that if a supervisor is trajectory model non-blocking, then it is also language model non-blocking. On the other hand, if a plant as well as a supervisor are deterministic, and the supervisor is language model non-blocking, then it is also trajectory model non-blocking.

Remark 4 It must be kept in mind that although trajectory model non-blocking is a stronger notion than language model non-blocking, it is possible to have a system which is trajectory model non-blocking yet it "blocks". Consider for example a simple NSM with event set $\Sigma = \{a\}$ and consisting of three states, in which there is nondeterministic transition on event a from the initial marked state to the other two states, only one of which is marked; no other transition is defined. Then the recognized as well as generated trajectory set of this NSM is given by $pr(dom(\emptyset(a, \{a\}))$. Consequently, it is trajectory model non-blocking. However, after executing the event a in its initial state, the system can reach an unmarked state from which no marked state is reachable.

Thus the notion of trajectory model non-blocking is not adequate when a "blocking" as well as a "non-blocking" state is reachable by the execution of the same set of refusal-traces. However, in a practical setting this is unlikely, as whenever a "blocking" and a "non-blocking" state is reached by the execution of the same *trace* due to the presence of nondeterminism, we expect the refusal-sets at the two states or the intermediate states to be different (as they are physically different states), so it is not possible to reach the two states by the same set of refusal-traces. ■

Next we obtain a necessary and sufficient condition for the existence of a non-marking and language model non-blocking supervisor. This result is then used to obtain a necessary and sufficient condition for the existence of a non-marking and trajectory model non-blocking supervisor.

Theorem 5 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$ with $A \cup B = \Sigma$, and $K^m \subseteq L((P^m)^{\Sigma - A})$ with $K^m \neq \emptyset$. Then there exists a non-marking and language model non-blocking supervisor with trajectory model (S, S) such that $L(P^m{}_A\|_B\, S) = K^m$ if and only if

Relative-closure: $pr(K^m) \cap L((P^m)^{\Sigma - A}) = K^m$
Controllability: $pr(K^m)(A - B) \cap L(P^{\Sigma - A}) \subseteq pr(K^m)$.

In this case, S can be chosen to be $det(pr(K^m))$.

Proof: We begin with the proof for necessity. Suppose there exists a non-marking and language model non-blocking supervisor with trajectory model (S, S) such that $L(P^m{}_A\|_B\, S) = K^m$. Then

$$pr(K^m) = pr(L(P^m{}_A\|_B\, S)) = L(P{}_A\|_B\, S), \tag{4}$$

where the last equality follows from the supervisor being language model non-blocking. It follows from Definitions 4 and 5 that $(P^m)^{\Sigma-A} \subseteq P^{\Sigma-A}$, which implies that $pr(K^m) \subseteq pr(L((P^m)^{\Sigma-A})) \subseteq L(P^{\Sigma-A})$. Hence it follows from the necessity part of [16, Theorem 4] that $pr(K^m)(A-B) \cap L(P^{\Sigma-A}) \subseteq pr(K^m)$–i.e., the controllability property is satisfied. It remains to show that the relative-closure property is also satisfied. We have the following series of equalities:

$$\begin{aligned} pr(K^m) \cap L((P^m)^{\Sigma-A}) &= L(P\,{}_A\|_B\, S) \cap L((P^m)^{\Sigma-A}) \\ &= L((P\,{}_A\|_B\, S)\,{}_\Sigma\|_A\, P^m) \\ &= L((P\,{}_A\|_A\, P^m)\,{}_A\|_B\, S) \\ &= L((P\,{}_A\|_A\, P^m)^{\Sigma-A}) \cap L(S^{\Sigma-B}), \end{aligned} \tag{5}$$

where the first equality follows from Equation 4, the second equality follows from Corollary 7 and Proposition 3, the third equality follows from associativity of PSC (Corollary 8), and the final equality follows from Corollary 7. On the other hand, we have

$$K^m = L(P^m\,{}_A\|_B\, S) = L((P^m)^{\Sigma-A}) \cap L(S^{\Sigma-B}), \tag{6}$$

where the last equality follows from Proposition 3 and Corollary 7. It follows from Equations 5 and 6 that in order to show that the relative-closure property is satisfied, it suffices to show $L((P\,{}_A\|_A\, P^m)^{\Sigma-A}) = L((P^m)^{\Sigma-A})$. We have

$$\begin{aligned} L((P\,{}_A\|_A\, P^m)^{\Sigma-A}) &= L((P\,{}_A\|_A\, P^m)\,{}_A\|_{\Sigma-A}\, det((\Sigma-A)^*)) \\ &= L(P\,{}_A\|_\Sigma\, (P^m\,{}_A\|_{\Sigma-A}\, det((\Sigma-A)^*))) \\ &= L(P\,{}_A\|_\Sigma\, (P^m)^{\Sigma-A}) \\ &= L(P^{\Sigma-A}) \cap L((P^m)^{\Sigma-A}) \\ &= L((P^m)^{\Sigma-A}), \end{aligned}$$

where the first and third equalities follow from the definition of augmentation, the second equality follows from the associativity of PSC (Corollary 8), and the fourth equality follows from Proposition 3. This completes the proof of necessity.

Next we prove sufficiency. Suppose the relative-closure and controllability properties are satisfied. Then it follows from the sufficiency part of [16, Theorem 4] that the non-marking deterministic supervisor with trajectory model (S, S), where $S := det(pr(K^m))$, yields

$$L(P\,{}_A\|_B\, S) = pr(K^m), \tag{7}$$

and S^m is arbitrary. We select $S^m = S$, so the supervisor is non-marking. It follows from Equation 7 that

$$pr(K^m) = L(P^{\Sigma-A}) \cap L(S^{\Sigma-B}). \tag{8}$$

Using the relative-closure property gives the following series of equalities:

$$\begin{aligned} K^m &= pr(K^m) \cap L((P^m)^{\Sigma-A}) \\ &= [L(P^{\Sigma-A}) \cap L(S^{\Sigma-B})] \cap L((P^m)^{\Sigma-A}) \\ &= L((P^m)^{\Sigma-A}) \cap L(S^{\Sigma-B}) \\ &= L(P^m\,{}_A\|_B\, S). \end{aligned}$$

Since $pr(K^m) = L(P\,_A\|_B\, S)$ and $K^m = L(P^m\,_A\|_B\, S)$, the supervisor is language model non-blocking. ■

Remark 5 In contrast to the standard conditions [14] for the existence of a non-marking and language model non-blocking supervisor in the absence of driven events, the controllability and relative-closure conditions given in Theorem 5 refer to the language model of the *augmented* plant. It is easily demonstrated [16, Example 2] that this language model depends on the *trajectory model* of the plant and generally cannot be determined if *only* the language model of the plant is known. ■

We now obtain necessary and sufficient conditions for the existence of a deterministic non-marking and *trajectory model non-blocking* supervisor that imposes a desired closed-loop recognized language. The following preliminary results are needed.

Lemma 1 [16, Lemma 6] Let P, Q be generated trajectory sets, $A, B \subseteq \Sigma$, $f_p, e_p \in P$, $f_q, e_q \in Q$, with $f_p \sqsubseteq e_p$, $f_q \sqsubseteq e_q$. Then $f_p\,_A\|_B\, f_q \subseteq e_p\,_A\|_B\, e_q$.

Lemma 2 Let (P^m, P) be a trajectory model, $\emptyset \neq H = pr(H) \subseteq \Sigma^*$, and $K := L(P) \cap H$. Then

1. $P\,_\Sigma\|_\Sigma\, det(H) = P\,_\Sigma\|_\Sigma\, det(K)$,
2. $P^m\,_\Sigma\|_\Sigma\, det(H) = P^m\,_\Sigma\|_\Sigma\, det(K)$.

Proof: By T3, every refusal-trace in a generated trajectory set is dominated by a saturated trajectory. By T5, every refusal-trace in a recognized trajectory set is dominated by a saturated marked trajectory. By Lemma 1, it suffices to show that for any saturated trajectories $e \in P$ (respectively, $e \in P^m$), $f \in det(H)$, $g \in det(K)$, we have

$$e\,_\Sigma\|_\Sigma\, f \subseteq P\,_\Sigma\|_\Sigma\, det(K) \quad \text{(respectively, } P^m\,_\Sigma\|_\Sigma\, det(K)) \tag{9}$$

$$e\,_\Sigma\|_\Sigma\, g \subseteq P\,_\Sigma\|_\Sigma\, det(H) \quad \text{(respectively, } P^m\,_\Sigma\|_\Sigma\, det(H)). \tag{10}$$

The left side of Equation 9 is empty unless $tr(e) = tr(f) \in K$, while the left side of Equation 10 is empty unless $tr(e) = tr(g) \in K$. Thus, to establish Equations 9 and 10, it suffices to show that for $e \in P_{sat}$ with $tr(e) := t \in K$, $f := trj_H(t)$, $g := trj_K(t)$, we have

$$e\,_\Sigma\|_\Sigma\, f = e\,_\Sigma\|_\Sigma\, g. \tag{11}$$

We have for each $k \leq |t|$, $\sigma \in \Sigma_k(f)$ (respectively, $\sigma \in \Sigma_k(g)$) if and only if $t^k\sigma \notin H$ (respectively, $t^k\sigma \notin K$). It follows from Definition 3 that

$$\begin{aligned} e\,_\Sigma\|_\Sigma\, f &= \{h \in 2^\Sigma(\Sigma \times 2^\Sigma)^* \mid tr(h) = t, \Sigma_k(h) \subseteq \Sigma_k(e) \cup \Sigma_k(f),\ k = 0, \ldots, |t|\}, \\ e\,_\Sigma\|_\Sigma\, g &= \{h \in 2^\Sigma(\Sigma \times 2^\Sigma)^* \mid tr(h) = t, \Sigma_k(h) \subseteq \Sigma_k(e) \cup \Sigma_k(g),\ k = 0, \ldots, |t|\}. \end{aligned}$$

Thus, to establish Equation 11, it suffices to show that

$$\Sigma_k(e) \cup \Sigma_k(f) = \Sigma_k(e) \cup \Sigma_k(g), \quad k = 0, \ldots, |t|. \tag{12}$$

Since $K \subseteq H$, it follows that $\Sigma_k(f) \subseteq \Sigma_k(g)$. On the other hand, if $\sigma \in \Sigma_k(g) - \Sigma_k(e)$, then $t^k\sigma \notin K$ and, since $e \in P_{sat}$, it follows that $e^k(\sigma, \emptyset) \in P$, so $t^k\sigma \in L(P)$. Hence, $t^k\sigma \notin H$, so $\sigma \in \Sigma_k(f)$ proving Equation 12. ∎

The following result can be proved using the above lemma and forms the basis for the existence of trajectory model non-blocking supervisor presented in the next theorem.

Proposition 4 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$ with $A \cup B = \Sigma$, and $K^m \subseteq L((P^m)^{\Sigma-A})$ with $K^m \neq \emptyset$. If (S, S) is any non-marking language model non-blocking deterministic supervisor with $L(P^m \,{}_A\|_B\, S) = K^m$, then

$$P \,{}_A\|_B\, det(pr(K^m)) = P \,{}_A\|_B\, S, \tag{13}$$
$$P^m \,{}_A\|_B\, det(pr(K^m)) = P^m \,{}_A\|_B\, S. \tag{14}$$

Proof: It follows that the relative-closure and controllability conditions in Theorem 3 hold, and hence that $(det(pr(K^m)), det(pr(K^m)))$ is a non-marking and language model non-blocking supervisor with $L(P^m \,{}_A\|_B\, det(pr(K^m))) = K^m$. Thus,

$$pr(K^m) = L(P \,{}_A\|_B\, det(pr(K^m))) = L(P^{\Sigma-A}) \cap L((det(pr(K^m)))^{\Sigma-B}).$$

This together with Lemma 2 gives

$$P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, (det(pr(K^m)))^{\Sigma-B} = P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, det(pr(K^m)), \tag{15}$$
$$(P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, (det(pr(K^m)))^{\Sigma-B} = (P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, det(pr(K^m)). \tag{16}$$

Since (S, S) is language model non-blocking,

$$pr(K^m) = L(P \,{}_A\|_B\, S) = L(P^{\Sigma-A}) \cap L(S^{\Sigma-B}).$$

Hence it follows from Lemma 2 and the fact that $S^{\Sigma-B}$ is deterministic that

$$P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, S^{\Sigma-B} = P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, det(pr(K^m)), \tag{17}$$
$$(P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, S^{\Sigma-B} = (P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, det(pr(K^m)). \tag{18}$$

From Equations 15-18 we have

$$P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, (det(pr(K^m)))^{\Sigma-B} = P^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, S^{\Sigma-B},$$
$$(P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, (det(pr(K^m)))^{\Sigma-B} = (P^m)^{\Sigma-A} \,{}_\Sigma\|_\Sigma\, S^{\Sigma-B}.$$

This implies Equations 13-14. ∎

Theorem 6 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$ with $A \cup B = \Sigma$, and $K^m \subseteq L((P^m)^{\Sigma - A})$ with $K^m \neq \emptyset$. Then there exists a non-marking and trajectory model non-blocking deterministic supervisor with trajectory model (S, S) such that $L(P^m{}_A\|_B\, S) = K^m$ if and only if

Relative-closure: $pr(K^m) \cap L((P^m)^{\Sigma - A}) = K^m$
Controllability: $pr(K^m)(A - B) \cap L(P^{\Sigma - A}) \subseteq pr(K^m)$
Trajectory-closure: $P_A\|_B\, det(pr(K^m)) = pr[P^m{}_A\|_B\, det(pr(K^m))]$.

In this case, S can be chosen to be $det(pr(K^m))$.

Proof: We first prove the sufficiency. Since the relative-closure and controllability conditions hold, it follows from the sufficiency part of Theorem 5 that the non-marking deterministic supervisor (S, S), where $S := det(pr(K^m))$, yields $L(P^m{}_A\|_B\, S) = K^m$ and $L(P_A\|_B\, S) = pr(K^m)$. It follows from the trajectory-closure condition that the supervisor is also trajectory model non-blocking.

Next we prove the necessity. Suppose there exists a non-marking and trajectory model non-blocking deterministic supervisor with trajectory model (S, S) such that $L(P^m{}_A\|_B\, S) = K^m$. Since a trajectory model non-blocking supervisor is also language model non-blocking, it follows from Theorem 5 that the relative-closure and controllability conditions hold, and from Proposition 4 that

$$P_A\|_B\, det(pr(K^m)) = P_A\|_B\, S, \tag{19}$$
$$P^m{}_A\|_B\, det(pr(K^m)) = P^m{}_A\|_B\, S. \tag{20}$$

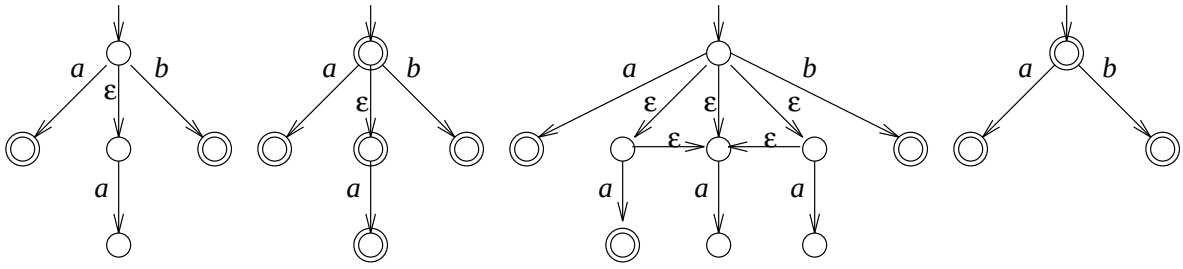
Since (S, S) is trajectory model non-blocking, we have $P_A\|_B\, S = pr(P^m{}_A\|_B\, S)$. Hence, the trajectory-closure condition is implied by Equations 19-20. ■

Remark 6 The controllability condition of Theorem 5 is needed for the existence of a supervisor such that the closed-loop generated language equals the closure of the language to be recognized by the closed-loop system, i.e., $L(P_A\|_B\, S) = pr(K^m)$. The relative-closure condition is needed so that the corresponding non-marking supervisor (S, S) yields equality of closed-loop recognized language and the desired language, i.e., $L(P^m{}_A\|_B\, S) = K^m$. Thus this design automatically yields that the non-marking supervisor (S, S) is also language model non-blocking. (The language model non-blocking property in a sense comes for free!) The extra trajectory-closure condition of Theorem 6 is needed so that the non-marking supervisor (S, S) is also trajectory model non-blocking. Thus all three conditions of Theorem 6 are needed for the existence of a non-marking and trajectory model non-blocking deterministic supervisor so that the closed-loop recognized language is the desired one. ■

It is important to understand some of the implications of Theorem 6. The three conditions of Theorem 6 are sufficient for the existence of a non-marking and trajectory model non-blocking supervisor that is *not necessarily deterministic* and imposes the desired marked language constraint. However, these conditions are not necessary for the existence of such a

supervisor; rather they are necessary for the existence of a supervisor that is also *deterministic*. Thus (i) the condition of determinism cannot be dropped in Theorem 6, and (ii) it is possible to obtain a nondeterministic supervisor meeting the other specifications of Theorem 6 even in the absence of the trajectory-closure condition (refer to the example below). I.e., it is possible to weaken the conditions of Theorem 6 by relaxing the requirement of determinism of the supervisor. However, deterministic supervisors may be preferred due to their ease in implementation.

Example 2 Consider the plant NSM defined on the event set $\Sigma = \{a, b\}$, shown in Figure 2(a). Since the refusal-trace $\{b\}(a, \{a, b\})$ belongs to the generated trajectory set of the plant, and there exists no refusal-trace in the recognized trajectory set of the plant with $\{b\}(a, \{a, b\})$ as its prefix, the plant is trajectory model *blocking*.



(a) plant/closed-loop system under deterministic supervisor (b) nondeterministic supervisor (c) closed-loop syatem under nondeterministic supervisor (d) deterministic supervisor

Figure 2: Diagram illustrating Example 2

Consider the nondeterministic supervisor depicted in Figure 2(b). The supervisor has the same structure as the plant except that all its states are marked, so that it is non-marking. Let the priority set of both plant and supervisor be the entire event set. The closed-loop system under the nondeterministic supervision is shown in Figure 2(c). The generated trajectory set of the closed-loop system is same as that of the plant. However, the recognized trajectory set is larger than that of the plant. In particular, the refusal-trace $\{b\}(a, \{a, b\})$ belongs to the recognized trajectory set of the closed-loop system, and the closed-loop system is trajectory model *non-blocking*.

The generated language of the closed-loop system is $pr(a + b)$. The non-marking deterministic supervisor shown in Figure 2(d) generates this language. The closed-loop system under the supervision of the deterministic supervisor has the same NSM as the plant, and hence it is trajectory model blocking. Thus the trajectory-closure condition of Theorem 6 does not hold even though a non-marking and trajectory model non-blocking supervisor exists for the given plant. ■

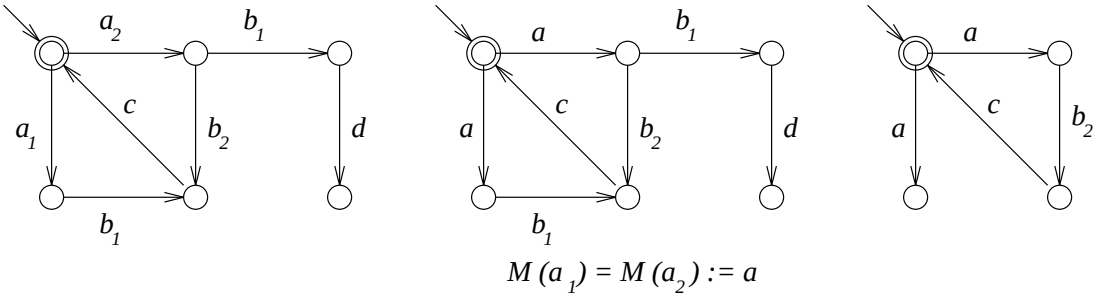
Remark 7 Theorem 6 requires that in order to determine the existence of a non-marking, trajectory model non-blocking and deterministic supervisor that imposes the desired recog-

nized language constraint, the conditions of controllability, relative-closure, and trajectory-closure must be verified. The conditions of controllability and relative-closure can be verified in polynomial time by simple extensions of the known tests for the deterministic setting [10] to the nondeterministic setting. We provide a simple test for verifying whether the trajectory-closure condition is satisfied. Given an NSM, it is said to be *co-accessible* if a marked state can be reached from every state that is reachable from the initial state. Let $\mathcal{P}$ be a NSM having (P^m, P) as its trajectory model, and $\mathcal{S}$ be a deterministic state machine with language model $(K^m, pr(K^m))$. Then it is easy to verify that the trajectory-closure condition holds whenever $\mathcal{P}\ {}_A\|_B\ \mathcal{S}$ is co-accessible. Note that co-accessibility of an NSM can be tested in time linear in the number of states in the NSM.

In case the conditions of Theorem 6 are not satisfied, it is possible to construct a *minimally restrictive supervisor* [14] that imposes the supremal sublanguage of K^m satisfying the three conditions as the desired behavior. In order to demonstrate the existence of such a sublanguage it suffices to show that the trajectory-closure condition is preserved under union, as it is known that the controllability and the relative-closure conditions are preserved under union. Since the PSC operation (with a fixed refusal-trace set, and fixed priority sets) as well as the prefix-closure operation over the space of *refusal-trace sets* are obtained by the extensions of the corresponding operations over the space of *refusal-traces*, they both distribute over arbitrary union. So it is readily obtained that the trajectory-closure condition is preserved under union. For a more formal treatment of existence and computation of minimally restrictive supervisors for nondeterministic systems with closed specification readers are referred to the recent article by Heymann-Lin [5] in which the nondeterministic system is first "converted" to a deterministic one by adding auxiliary transitions and states and then the traditional algorithms for the deterministic setting are applied for supervisory existence and computation. ■

6 Example

In this section, we illustrate some of the issues addressed in this paper. Figure 3(a)



(a) deterministic plant (b) nondeterministic plant (c) closed-loop system

Figure 3: Diagram illustrating the example

gives a deterministic model for a plant in which parts arrive at a machine from a conveyor

and are then processed. The incoming parts are of two types that differ slightly in their widths. The standard width is the wider one. Events a_1 and a_2 denote the arrival at the machine of wide and narrow parts respectively. Events b_1 and b_2 denote the input into the machine of a part with the guides set to wide and narrow respectively. The default setting of the guides is wide, but intervention by a controller can reset them to narrow. A wide part can only be input with the guides set to wide. A narrow part can be input with either guide setting. However, input of a narrow part with the guides set to wide leads to the machine jamming–event d. If a part is input with the correct guide setting, then it can be successfully processed and output–event c. It is assumed that a_1, a_2, c, d are uncontrollable events and that there is no sensor that can distinguish between the two widths of incoming parts–i.e., the observation mask $M(\cdot)$ identifies a_1 and a_2–say $M(a_1) = M(a_2) := a$. A natural control specification is that the supervised plant be non-blocking since this guarantees that continuous operation is possible.

It is clear that the performance specification cannot be met by any supervisor S of the Ramadge-Wonham type that is consistent with the observation mask. To prevent blocking arising from the uncontrollable jamming event d, S would need to disable b_1 following any occurrence of a_2. However, since the mask cannot distinguish between a_1 and a_2, S would also disable b_1 following any occurrence of a_1. But this would give a controlled plant that would get blocked with the arrival of the first wide part.

Suppose we replace the event labels a_1 and a_2 by their common mask value a, thereby obtaining the nondeterministic system shown in Figure 3(b). *By so identifying a_1 and a_2, the events are made indistinguishable from the viewpoints of specification, control and observation–whereas in the partially observed deterministic model, they are indistinguishable only from the viewpoint of observation.* For this system, however, the nondeterministic model is essentially equivalent to the partially observed one from the viewpoint of control since a_1, a_2 are uncontrollable and hence could not be distinguished in a supervisory control law. However, the non-blocking specification implicitly distinguishes between a_1 and a_2 and consequently forces the definition of a new type of non-blocking appropriate for nondeterministic systems.

Let $\mathcal{P}$ denote the nondeterministic state machine (NSM) depicted in Figure 3(b), and let $L(\mathcal{P}),\ L^m(\mathcal{P})$ denote its generated and recognized languages respectively. Then

$$L^m(\mathcal{P}) = [a(b_1 + b_2)c]^*, \qquad L(\mathcal{P}) = pr[[a(b_1 + b_2)c]^* ab_1 d].$$

Having replaced the original partially observed deterministic model with a completely observed nondeterministic model, let us consider whether the specification can be met by a supervisor of the Ramadge-Wonham type. The closed-loop nondeterministic system $\mathcal{Q}$ obtained by disabling b_1 following any occurrence of a is depicted in Figure 3(c). Since $L(\mathcal{Q}) = pr((ab_2c)^*) = pr(L^m(\mathcal{Q}))$, the supervisor is non-blocking from the language model point of view. However, this control design is clearly unsatisfactory since the closed-loop system can get blocked as can be seen from Figure 3(c). After all, the nondeterministic plant model is derived from the partially observed deterministic plant model, and there is no non-blocking Ramadge-Wonham type supervisor for that model.

The problem is that the usual language model definition of non-blocking given by $L(\mathcal{P}) = pr(L^m(\mathcal{P}))$ is not suitable for control specifications in a nondeterministic setting. This motivates us to consider the stronger non-blocking requirement of trajectory model non-blocking. Using trajectory models for the plant and supervisor, and PSC as the mode of interconnection, it is possible to design a supervisor so that the closed-loop system meets the stronger non-blocking requirement as described below.

For the open-loop NSM $\mathcal{P}$ depicted in Figure 3(b) it is given that a, c, d are uncontrollable. We regard b_1, the setting of guide to wide, as a controllable event–i.e., requiring the participation of both the plant and supervisor. However, we regard b_2 as a driven event. This models the possibility that the supervisor may request the input of a part with the guides set to narrow, but that this request may be refused by the plant if the arriving part is wide. Thus, for PSC-based design, the priority sets of the plant and supervisor are $A = \{a, b_1, c, d\}$ and $B = \{b_1, b_2\}$ respectively.

We indicated above that a language model non-blocking Ramadge-Wonham type supervisor could be constructed that gives $K_1^m := (ab_2c)^*$ as the closed-loop recognized language. We also noted that the closed-loop system is unsatisfactory since blocking can occur. Let us determine whether a PSC-based supervisor can impose the specification K_1^m without permitting blocking. The augmented plant $\mathcal{P}^{\Sigma-A}$ is shown in Figure 4(a) and has generated language given by $L(\mathcal{P}^{\Sigma-A}) = pr[[b_2^*a(b_2^*b_1 + b_2)b_2^*c]^*b_2^*ab_1b_2^*db_2^*]$. It is straightforward to verify

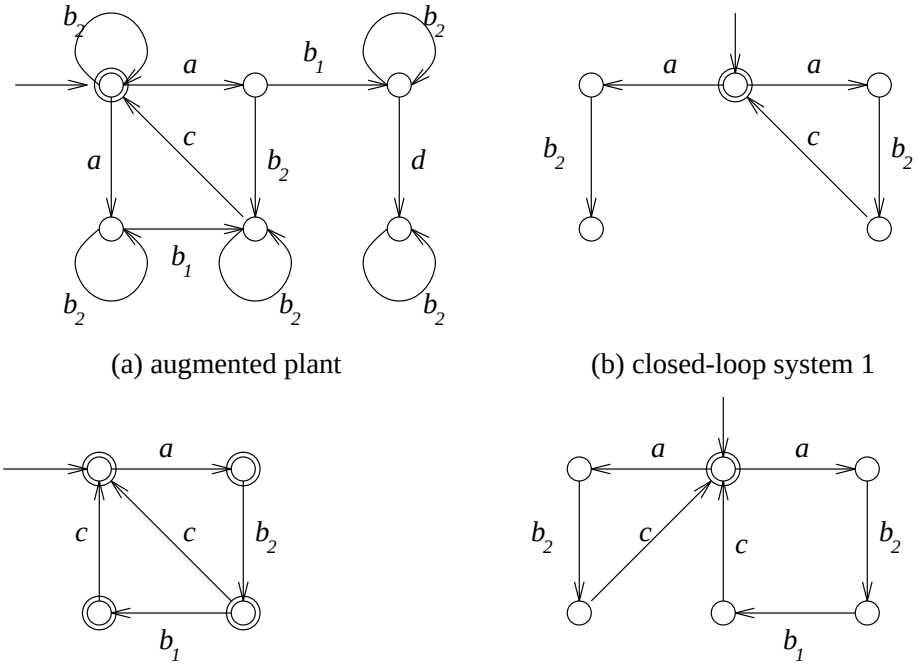


Figure 4: Diagram illustrating Example 3

that K_1^m satisfies the controllability and relative-closure conditions of Theorem 5. Thus, it

follows from Theorem 5 that the non-marking supervisor (S_1, S_1), $S_1 := det(pr(K_1^m))$, is language model non-blocking and imposes K_1^m as the closed-loop recognized language.

To determine whether there is any deterministic non-marking and *trajectory model non-blocking* supervisor that can impose K_1^m, we must check the trajectory-closure condition in Theorem 6. Let $\mathcal{S}_1$ denote the minimal deterministic state machine (with 3 states) with $L^m(\mathcal{S}_1) = L(\mathcal{S}_1) = pr(K_1^m)$. Let $\mathcal{Q}_1 := \mathcal{P} \,{}_A\|_B\, \mathcal{S}_1$ depicted in Figure 4(b). Letting (P^m, P) denote the trajectory model of $\mathcal{P}$, it follows from Theorem 3 that

$$P \,{}_A\|_B\, det(pr(K_1^m)) = T(\mathcal{Q}_1), \quad P^m \,{}_A\|_B\, det(pr(K_1^m)) = T^m(\mathcal{Q}_1).$$

Since $e := \emptyset(a, \emptyset)(b_2, \{c\}) \in T(\mathcal{Q}_1) - pr(T^m(\mathcal{Q}_1))$, the trajectory-closure condition fails to hold. Thus, K_1^m cannot be imposed without trajectory model blocking.

Consider the alternative specification $K_2^m := (ab_2(\epsilon + b_1)c)^*$, which also satisfies the controllability and relative-closure conditions of Theorem 5. Let $\mathcal{S}_2$ denote the deterministic state machine with $L^m(\mathcal{S}_2) = L(\mathcal{S}_2) = pr(K_2^m)$ depicted in Figure 4(c). The resulting closed-loop system $\mathcal{Q}_2 := P \,{}_A\|_B\, \mathcal{S}_2$ is shown in Figure 4(d). Since $T(\mathcal{Q}_2) = pr(T^m(\mathcal{Q}_2))$, the trajectory-closure condition holds. Thus, the non-marking deterministic supervisor (S_2, S_2), $S_2 := det(pr(K_2^m))$, is trajectory model non-blocking and imposes K_2^m as the closed-loop recognized language.

The supervisor implements the following control strategy: When a part arrives, the supervisor requests that it be input with the guides set to narrow. If the part happens to be narrow, the plant accepts this request and executes the event b_2. However, if the part is wide, the plant refuses to execute the event and the transition on b_2 occurs only in the supervisor. The part is then input with the guides set to wide. By following this strategy, a narrow part is never input with the guides set to wide, so jamming does not occur. Allowing for the possibility that a supervisor-initiated event may be refused by the plant is an essential feature of this control design.

7 Conclusion

In this paper, we have extended our earlier work on supervisory control of nondeterministic systems to include markings so that the issue of blocking can be investigated. Since language model non-blocking is inadequate for certain nondeterministic systems, the stronger requirement of trajectory model non-blocking is introduced. Necessary and sufficient conditions for the existence of language model non-blocking as well as trajectory model non-blocking supervisors that meet given language specifications are obtained in terms of standard controllability and relative-closure condition, and a new condition called the trajectory-closure condition. The co-accessibility of a certain NSM, which is polynomially testable, can be used to verify the trajectory-closure condition. The trajectory-closure condition is preserved under union, so it is possible to obtain a minimally restrictive supervisor.

A Corollaries of Theorems 3 and 4

The first part of the following corollary states that the trajectory model contains enough detail to support the prioritized synchronous composition of NSM's. The second part of Corollary 4 states that the trajectory model serves as a *language congruence* [4] with respect to the operation of prioritized synchronous composition. This fact was first mentioned in [6] without proof.

Corollary 4 Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2$ be NSM's such that $T^m(\mathcal{P}_1) = T^m(\mathcal{P}_2), T(\mathcal{P}_1) = T(\mathcal{P}_2)$, $T^m(\mathcal{Q}_1) = T^m(\mathcal{Q}_2), T(\mathcal{Q}_1) = T(\mathcal{Q}_2)$. Then for any $A, B \subseteq \Sigma$:

1. $T^m(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = T^m(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2)$ and $T(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = T(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2)$,
2. $L^m(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = L^m(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2)$ and $L(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = L(\mathcal{P}_2 \,{}_A\|_B\, \mathcal{Q}_2)$.

Proof: The first part follows immediately from Theorem 3. The second part follows from the first part since $L^m(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1) = L(T^m(\mathcal{P}_1 \,{}_A\|_B\, \mathcal{Q}_1))$, etc. ■

The following corollary was first reported in [6, Theorem 13.3]. However, its rigorous demonstration follows from the precise definition of PSC of refusal-traces given by us in [16, Definition 10] and the development of Section 3, in particular Theorem 3.

Corollary 5 Let (P^m, P) and (Q^m, Q) be two trajectory models, and $A, B \subseteq \Sigma$. Then $(P^m \,{}_A\|_B\, Q^m, P \,{}_A\|_B\, Q)$ is a trajectory model.

Proof: By hypothesis, there exist NSM's $\mathcal{P}$ and $\mathcal{Q}$ such that $(T^m(\mathcal{P}), T(\mathcal{P})) = (P^m, P)$ and $(T^m(\mathcal{Q}), T(\mathcal{Q})) = (Q^m, Q)$. It follows from Theorem 3 that $(P^m \,{}_A\|_B\, Q^m, P \,{}_A\|_B\, Q) = (T^m(\mathcal{P} \,{}_A\|_B\, \mathcal{Q}), T(\mathcal{P} \,{}_A\|_B\, \mathcal{Q}))$, which completes the proof. ■

The following corollary states that when the priority sets of two systems are each equal to the entire event set, then the language model of the composed system can be obtained as the intersection of the individual language models. Thus if two systems operate in strict synchrony, then the language model of the composed system can be determined from the language models of the individual systems (the trajectory models of the individual systems may not be known).

Corollary 6 Let $\mathcal{P} := (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P}, X^m_\mathcal{P}), \mathcal{Q} := (X_\mathcal{Q}, \Sigma, \delta_\mathcal{Q}, x^0_\mathcal{Q}, X^m_\mathcal{Q})$ be NSM's. Then

1. $L^m(\mathcal{P} \,{}_\Sigma\|_\Sigma\, \mathcal{Q}) = L^m(\mathcal{P}) \cap L^m(\mathcal{Q})$,
2. $L(\mathcal{P} \,{}_\Sigma\|_\Sigma\, \mathcal{Q}) = L(\mathcal{P}) \cap L(\mathcal{Q})$.

Proof: We only prove the first part; the second part can be proved analogously. Let $\mathcal{R} := \mathcal{P} \,{}_\Sigma\|_\Sigma\, \mathcal{Q}$. We have $L^m(\mathcal{P}) = L(T^m(\mathcal{P})), L^m(\mathcal{Q}) = L(T^m(\mathcal{Q})), L^m(\mathcal{R}) = L(T^m(\mathcal{R}))$, and it follows from Theorem 3 that $T^m(\mathcal{R}) = T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})$. Hence it suffices to show that $L(T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})) = L(T^m(\mathcal{P})) \cap L(T^m(\mathcal{Q}))$. Pick $s \in L(T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q}))$. Then there exists $e \in T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})$ such that $tr(e) = s$. Since $e \in T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})$, it

follows that there exist $e_p \in T^m(\mathcal{P}), e_q \in T^m(\mathcal{Q})$ such that $e \in e_p \,{}_A\|_B\, e_q$ and $tr(e_p) = tr(e_q) = tr(e) = s$. (Refer to [16, Remark 6].) I.e., $s \in L(T^m(\mathcal{P})) \cap L(T^m(\mathcal{Q}))$. This proves that $L(T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})) \subseteq L(T^m(\mathcal{P})) \cap L(T^m(\mathcal{Q}))$.

Similarly, given $s \in L(T^m(\mathcal{P})) \cap L(T^m(\mathcal{Q}))$, there exist $e_p \in T^m(\mathcal{P}), e_q \in T^m(\mathcal{Q})$ such that $tr(e_p) = tr(e_q) = s$. Choose $e \in e_p \,{}_\Sigma\|_\Sigma\, e_q$, which is clearly nonempty. Then $e \in T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q})$ and $tr(e) = s$. (Refer to [16, Remark 6].) Consequently, we have $s \in L(T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q}))$, which proves that $L(T^m(\mathcal{P})) \cap L(T^m(\mathcal{Q})) \subseteq L(T^m(\mathcal{P}) \,{}_\Sigma\|_\Sigma\, T^m(\mathcal{Q}))$. ■

Corollary 7 Let (P^m, P) and (Q^m, Q) be trajectory models. Then

1. $L(P^m \,{}_\Sigma\|_\Sigma\, Q^m) = L(P^m) \cap L(Q^m)$,
2. $L(P \,{}_\Sigma\|_\Sigma\, Q) = L(P) \cap L(Q)$.

Proof: From the hypothesis there exist NSM's $\mathcal{P}, \mathcal{Q}$ such that $(T^m(\mathcal{P}), T(\mathcal{P})) = (P^m, P)$ and $(T^m(\mathcal{Q}), T(\mathcal{Q})) = (Q^m, Q)$. Hence the result follows from Corollary 6. ■

The following corollary is an immediate consequence of Theorem 4. Part 2 of this corollary was stated without proof in [6]; a direct proof of part 2 that does not depend on the corresponding result for NSM's is given in [16, Theorem 3].

Corollary 8 Let (P^m, P), (Q^m, Q) and (R^m, R) be trajectory models and $A, B, C \subseteq \Sigma$. Then

1. $(P^m \,{}_A\|_B\, Q^m) \,{}_{A\cup B}\|_C\, R^m = P^m \,{}_A\|_{B\cup C}\, (Q^m \,{}_B\|_C\, R^m)$,
2. $(P \,{}_A\|_B\, Q) \,{}_{A\cup B}\|_C\, R = P \,{}_A\|_{B\cup C}\, (Q \,{}_B\|_C\, R)$.

Proof: From the hypothesis, there exist NSM's $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ such that $(T^m(\mathcal{P}), T(\mathcal{P})) = (P^m, P)$, $(T^m(\mathcal{Q}), T(\mathcal{Q})) = (Q^m, Q)$, and $(T^m(\mathcal{R}), T(\mathcal{R})) = (R^m, R)$. It is easily shown using the result of Theorem 3 that

$$\begin{aligned}
(P^m \,{}_A\|_B\, Q^m) \,{}_{A\cup B}\|_C\, R^m &= T^m((\mathcal{P} \,{}_A\|_B\, \mathcal{Q}) \,{}_{A\cup B}\|_C\, \mathcal{R}),\\
P^m \,{}_A\|_{B\cup C}\, (Q^m \,{}_B\|_C\, R^m) &= T^m(\mathcal{P} \,{}_A\|_{B\cup C}\, (\mathcal{Q} \,{}_B\|_C\, \mathcal{R})),\\
(P \,{}_A\|_B\, Q) \,{}_{A\cup B}\|_C\, R &= T((\mathcal{P} \,{}_A\|_B\, \mathcal{Q}) \,{}_{A\cup B}\|_C\, \mathcal{R}),\\
P \,{}_A\|_{B\cup C}\, (Q \,{}_B\|_C\, R) &= T(\mathcal{P} \,{}_A\|_{B\cup C}\, (\mathcal{Q} \,{}_B\|_C\, \mathcal{R})).
\end{aligned}$$

Thus the result follows from Theorem 4. ■

References

[1] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. Wong-Toi, and G. F. Franklin. Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, 38(7):1040–1059, July 1993.

[2] B. A. Brandin and W. M. Wonham. Supervisory control of timed discrete event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, February 1994.

[3] C. H. Golaszewski and P. J. Ramadge. Control of discrete event processes with forced events. In *Proceedings of 26th IEEE Conference on Decision and Control*, pages 247–251, Los Angeles, CA, 1987.

[4] M. Heymann. Concurrency and discrete event control. *IEEE Control Systems Magazine*, 10(4):103–112, 1990.

[5] M. Heymann and F. Lin. On observability and nondeterminism in discrete event control. In *Proceedings of 1995 Annual Allerton Conference*, Urbana, IL, October 1995. To appear.

[6] M. Heymann and G. Meyer. Algebra of discrete event processes. Technical Report NASA 102848, NASA Ames Research Center, Moffett Field, CA, June 1991.

[7] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1985.

[8] K. Inan. An algebraic approach to supervisory control. *Mathematics of Control, Signals and Systems*, 5:151–164, 1992.

[9] K. Inan and P. Varaiya. Algebras of discrete event models. *Proceedings of the IEEE*, 77(1):24–38, 1989.

[10] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, Boston, MA, 1995.

[11] R. Kumar, V. K. Garg, and S. I. Marcus. On controllability and normality of discrete event dynamical systems. *Systems and Control Letters*, 17(3):157–168, 1991.

[12] R. Milner. *A Calculus of Communicating Systems*. Springer Verlag, 1980.

[13] I. Phillips. Refusal testing. *Theoretical Computer Science*, 50:241–284, 1987.

[14] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[15] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of IEEE: Special Issue on Discrete Event Systems*, 77:81–98, 1989.

[16] M. A. Shayman and R. Kumar. Supervisory control of nondeterministic systems with driven events via prioritized synchronization and trajectory models. *SIAM Journal of Control and Optimization*, 33(2):469–497, March 1995.

[17] J. G. Thistle. Logical aspects of control of discrete event systems: a survey of tools and techniques. In Guy Cohen and Jean-Pierre Quadrat, editors, *Lecture Notes in Control and Information Sciences 199*, pages 3–15. Springer-Verlag, New York, 1994.