Centralized and Decentralized Supervisory Control of Nondeterministic Systems Under Partial Observation [1] [2]

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 of Systems Research
University of Maryland
College Park, MD 20742
Email: shayman@src.umd.edu

November 20, 2005

[1] Initial version of this paper has appeared as [18].

[2] This research was supported in part by the Center for Robotics and Manufacturing, University of Kentucky, and in part by the National Science Foundation under grants CDR-8803012, EEC-94-02384, ECS-9312587 and ECS-9409712.

Abstract

In this paper we extend our earlier work on supervisory control of nondeterministic systems using prioritized synchronization as the mechanism of control and trajectory model as the modeling formalism by considering design of supervisors under partial observation. We introduce the notion of *observation-compatible* systems and show that prioritized synchronous composition of observation-compatible systems can be used as a mechanism of control of nondeterministic systems under partial observation in presence of driven events. Necessary and sufficient conditions, that depend on the trajectory model as opposed to the language model of the plant, are obtained for the existence of *centralized* as well as *decentralized* supervision. Our work on centralized control shows that the results of the traditional supervisory control can be "extended" to the above setting provided the supervisor is deterministic and the observation mask is projection type. On the other hand, our work on decentralized control is based on a new relation between controllability, observability, co-observability, and PSC that we derive in this paper.

Keywords: discrete event systems, supervisory control, partial observation, nondeterministic automata, driven events, prioritized synchronization, trajectory models, controllability, observability, co-observability

AMS (MOS) subject classifications: 68Q75, 93B25, 93C83

1 Introduction

Discrete event systems (DES's) are systems which involve quantities that take a discrete set of values and which evolve according to occurrence of certain discrete qualitative changes, called *events*, such as arrival of a customer in a queue, termination of an algorithm in a computer program, loss of a message packet in a communication network, breakdown of a machine in a manufacturing system, etc. The theory of supervisory control of DES's was introduced by Ramadge and Wonham [26, 27] for designing controllers so that the controlled system satisfies certain desired *qualitative* constraints, such as a buffer in a manufacturing system should never overflow, a message sequence in a communication network must be received in the same order as it was transmitted, etc.

Such qualitative behavior of a *deterministic*[1] DES can be described by the set of all possible event *traces*, called a *language* model, that the system can execute starting from its initial state. However, due to partial observation and/or unmodeled dynamics, it is too restrictive to require determinism of a system. If a DES is nondeterministic, then its language model may not adequately describe its qualitative behavior, and more detailed models are needed. Several models such as *failures model* [10], *refusal-trace model* [25], *ready-trace model* [1], *bisimulation model* [23, 24], etc., have been proposed in the literature for representing qualitative behavior of nondeterministic DES's. A nice comparative study of such modeling formalisms can be found in [2, 31]. As a designer, it is desirable to choose the *least detailed* modeling formalism that is adequate for the design task at hand. As is argued below, this is the reason for us to choose the *trajectory model* proposed by Heymann [8], also known as *refusal-trace model*, for representing nondeterministic DESs.

Most of the prior work on supervisory control of DES's such as [26, 16, 4] essentially use *strict synchronous composition* (SSC) of *plant* DES and *supervisor* DES as the mechanism of control. In SSC of systems, it is required that the common events must occur synchronously. This is restrictive, as due to nondeterminism the plant state is not uniquely known following the execution of a certain observed trace, and the set of executable events in each such state may differ. If we require strict synchronization, then the supervisor is restricted to enable those events that are executable in each of those states, which imposes a severe restriction on the supervisor. Moreover, there is no a priori reason for a supervisor to synchronously execute all the *uncontrollable* events that the plant can execute. Similarly, it is restrictive to require that the plant synchronously executes the so called *forcible* [7], or *command* [4], or *driven* [8] events that are initiated by the supervisor. The motivating example in [30, Section 2, Example 5] describes a nondeterministic plant that can be controlled only when the requirement of strict synchronization is relaxed.

In this paper we study the control of qualitative behavior of nondeterministic DES's using *prioritized synchronous composition* (PSC) as the mechanism of control. PSC was originally proposed by Heymann [8, 9] and was later applied for supervisory control in the deterministic setting by Balemi [3] and in the nondeterministic setting by Shayman-Kumar

[1] A DES is said to be deterministic if given the current state and an event that occurs in that state, the next state is uniquely determined.

[30]. PSC is a generalization of the SSC. The parallel operator considered by Inan [12, 13], an extension of the parallel operator defined in [14, 15], can be viewed as a generalization of PSC when applied to the so-called *improper* systems. However, while studying supervisory control only proper systems are considered; consequently the resulting operation is that of strict synchronization.

In PSC each system is associated with a certain priority set of events, and for an event to occur in the composition of a pair of systems operating in prioritized synchrony, each system having the priority over the event must participate. So if an event belongs to the common priority set, then it occurs synchronously. On the other hand, if a certain event belongs to the priority set of a single system, then it can occur asynchronously without the participation of the second system. However, the second system will participate whenever possible; such synchronization is called *broadcast synchronization*. Thus PSC does not impose the unnecessarily restrictive requirement of SSC that common events must always occur in synchrony. For supervisory control, the priority set of a plant consists of the uncontrollable and the controllable events, while the priority set of a supervisor consists of the controllable and the driven events. Since controllable events are in the priority sets of plant as well as supervisor, they always occur in synchrony in the controlled system, whereas the uncontrollable and the driven events may occur asynchronously.

Heymann showed via an example [8, Example 7] that if PSC is *admitted* as a mechanism of interconnection, then a modeling formalism which is more detailed than the failures model (and consequently, more detailed than the language model) is needed to adequately describe the behavior of nondeterministic DES's. For this reason, Heymann proposed the modeling formalism called *trajectory model*. A trajectory model consists of *generated* and *recognized* trajectories, also called *refusal-traces*, of a system. A refusal-trace is a sequence of alternating refusal sets and events, where a refusal set consists of those events that the system "refuses" to execute when offered at a certain execution point. Trajectory model is quite similar to the refusal-testing model of Phillips [25], but differs in its treatment of *silent* or epsilon transitions.

In our previous work [30, 19] we showed that the trajectory model can be used for adequately describing behaviors of nondeterministic DES's that may be interconnected using PSC, and showed that the operation of PSC is associative. 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 can be reduced to the operation of SSC provided the priority sets of the two systems exhaust the entire event set. Under this condition, we proved in [30, 19] that the PSC of a pair of systems is equivalent to SSC of appropriately augmented systems. In particular, if the plant is augmented with driven events and the supervisor is augmented with uncontrollable events, then the PSC of plant and supervisor is equivalent to SSC of augmented plant and augmented supervisor. Using these results we obtained necessary and sufficient conditions for the existence of a supervisor so that the language of the controlled plant equals a desired language.

In this paper we extend our earlier work on supervisory control of nondeterministic systems using prioritized synchronization as the mechanism of control and trajectory model

as the modeling formalism by considering design of supervisors under partial observation. Partial observation in the setting of supervisory control arises due to lack of sufficient number of sensors. As in the work of Lin and Wonham [21], we use a projection function, also called an *observation mask*, to represent such partial observation. A supervisor under partial observation must take identical control action following indistinguishable traces. We call this property of a supervisor *observation-compatibility*, which captures physically realizable supervisors. Such supervisors make control decisions based on *only* the observed event trace of the system, and do not require any "special" internal knowledge of the system.

We define the notion of observation-compatibility of a trajectory model and prove that this property is preserved under augmentation whenever the system is deterministic. Using this result we obtain a necessary and sufficient condition for the existence of an observation-compatible supervisor so that the language of the plant operating in prioritized synchrony with the supervisor equals the desired one. This result is then applied to obtain a supervisor which achieves mutually exclusive usage of a shared channel in a communication system. We also obtain conditions for the existence of *non-blocking* supervisors [27, 5].

Finally, we study the problem of decentralized supervision [29, 20, 22, 6, 32]. Decentralized supervision is inevitable when the plant is physically distributed for example as in communication networks and manufacturing systems. A supervisor is installed at each location of the "sub-plant". In such a situation, a supervisor is able to control a certain set of events, called *local* events, and is able to observe a partial set of events. The problem of decentralized supervision requires design of supervisors that are observation-compatible with respect to their own observation function, and control events in their own priority sets. This problem is naturally formulated in our framework. We show that the condition of controllability together with the condition of *co-observability* is necessary and sufficient for decentralized supervision. Our constructive proof is novel and is based on a nice relationship betweem controllability, observability, co-observability, and PSC that we derive in this paper. These conditions, however, are significantly different from the standard ones [21, 29], as they depend on the trajectory model (rather than language model) of the plant.

The remainder of the paper is organized as follows: In Section 2 we introduce the relevant notation. In Section 3, we define the notion of observation-compatibility and study some of its properties. In Section 4 we study the supervisory control problem under partial observation in the proposed framework and apply it for achieving mutually exclusive usage of a shared communication channel in a communication system. In Section 5 we study the problem of decentralized supervision. Finally, Section 6 concludes the work presented here.

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 as ϵ. A subset of Σ^* is called a language. Symbols H, K, etc., are used to denote languages. For a language $K \subseteq \Sigma^*$, the notation $pr(K) \subseteq \Sigma^*$, called the *prefix-closure* of K, is the set of all prefixes of traces from K. K is said to be prefix-closed if $K = pr(K)$.

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 [9, 30] of the type:

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

where $n \in \mathcal{N}$. The sequence $\sigma_1 \ldots \sigma_n \in \Sigma^*$ is the trace, and for each $i \leq n$, $\Sigma_i \subseteq \Sigma$ is a 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 $e \in 2^\Sigma(\Sigma \times 2^\Sigma)^*$, 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.,

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

The *trace* of e, denoted $tr(e) \in \Sigma^*$, is defined as $tr(e) := \sigma_1(e) \ldots \sigma_{|e|}(e)$. 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.

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 by $f \leq e$. For each $k \leq |e|$, the notation $e^k \leq e$ is used to denote the prefix of length k of e. The prefix-closure of e, denoted $pr(e) \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, is the set of all prefixes 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 by $f \sqsubseteq e$. The *dominance-closure* of e, denoted $dom(e) \subseteq 2^\Sigma(\Sigma \times 2^\Sigma)^*$, is the set of all refusal-traces dominated by e.

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 sequence of silent transitions.

The *ϵ-closure* of $x \in X_\mathcal{P}$, denoted $\epsilon^*_\mathcal{P}(x) \subseteq X_\mathcal{P}$, is defined inductively as:

$$x \in \epsilon^*_\mathcal{P}(x), \text{ and } x' \in \epsilon^*_\mathcal{P}(x) \Rightarrow \delta_\mathcal{P}(x', \epsilon) \subseteq \epsilon^*_\mathcal{P}(x),$$

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

[2] ϵ represents both an *internal* or *unobservable* event and an *internal* or *nondeterministic* choice [10, 23].

ϵ-closure of x. Using the definitions of the ϵ-closure and refusal maps, the transition function $\delta_{\mathcal{P}} : X_{\mathcal{P}} \times (\Sigma \cup \{\epsilon\}) \to 2^{X_{\mathcal{P}}}$ is extended (i) to the set of *traces*, denoted as $\delta^*_{\mathcal{P}} : X_{\mathcal{P}} \times \Sigma^* \to 2^{X_{\mathcal{P}}}$, which is defined in the usual way [11], and (ii) to the set of *refusal-traces*, denoted as $\delta^T_{\mathcal{P}} : X \times (2^{\Sigma}(\Sigma \times 2^{\Sigma})^*) \to 2^{X_{\mathcal{P}}}$, which is defined inductively as:

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

These maps are then used to obtain the language models and the trajectory models of $\mathcal{P}$ as follows:

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

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

$L(\mathcal{P}), L^m(\mathcal{P}), T(\mathcal{P}), T^m(\mathcal{P})$ are called the *generated language, recognized language, generated trajectory set, recognized trajectory set,* respectively, of $\mathcal{P}$. 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^m_1, K_1) and (K^m_2, K_2), are said to be equal, written as $(K^m_1, K_1) = (K^m_2, K_2)$, if $K^m_1 = K^m_2$ and $K_1 = K_2$; equality of two trajectory models is defined analogously.

Given a trajectory model, the trace map can be used to obtain the associated language model. On the other hand, 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 *deterministic* trajectory model[3] having the language model (K^m, K), which is defined as follows:

$$\begin{array}{rcl} trj_K(s) & := & \Sigma_0(s)(\sigma_1(s), \Sigma_1(s)) \ldots (\sigma_{|s|}(s), \Sigma_{|s|}(s)) \in 2^{\Sigma}(\Sigma \times 2^{\Sigma})^*, \text{ where} \\ \Sigma_k(s) & := & \{\sigma \in \Sigma \mid s^k\sigma \notin K\}, \forall k \leq |s|. \end{array}$$

Define $(det^m(K^m, K), det(K)) := (dom(trj_K(K^m)), dom(trj_K(K)))$. Then it is shown in [19, Proposition 1] that it is the unique deterministic trajectory model that has the language model (K^m, K).

In [8, 9, 30, 19] prioritized synchronous composition (PSC) of systems is used as the mechanism of control. In this setting, associated with each system is a priority set of events, which endows the system with the ability to prevent the occurrence of events belonging to its priority set; a system must participate in the execution of an event belonging to its priority set for that event to occur in the PSC with other system(s). Letting $\mathcal{P} \;{}_A\|_B\; \mathcal{Q}$ denote the PSC of NSM's $\mathcal{P}$ and $\mathcal{Q}$ with priority sets $A, B \subseteq \Sigma$ respectively, and $T^m(\mathcal{P}) \;{}_A\|_B\; T^m(\mathcal{Q})$, $T(\mathcal{P}) \;{}_A\|_B\; T(\mathcal{Q})$ denote the PSC of corresponding trajectory models, it was proved in [30, Theorem 2] and [19, Theorem 2] that

$$T^m(\mathcal{P}) \;{}_A\|_B\; T^m(\mathcal{Q}) = T^m(\mathcal{P} \;{}_A\|_B\; \mathcal{Q}); \quad T(\mathcal{P}) \;{}_A\|_B\; T(\mathcal{Q}) = T(\mathcal{P} \;{}_A\|_B\; \mathcal{Q}).$$

[3] A trajectory model (P^m, P) is said to be deterministic if there exists a deterministic state machine $\mathcal{P}$ such that $(T^m(\mathcal{P}), T(P)) = (P^m, P)$.

Various properties of PSC of trajectory models were studied in [30, 19]. In particular, associativity of PSC was proved [19, Proposition 2, Corollary 6], a language intersection result for the case when $A = B = \Sigma$ was obtained [19, Corollary 4, Corollary 5], and the notion of *augmentation* and its properties were studied.

We recall from [30, 19] that the *augmentation* of an NSM $\mathcal{P}$ by an event set $D \subset \Sigma$ is the NSM $\mathcal{P}^D := \mathcal{P} \parallel_\emptyset \mathcal{D}$, where $\mathcal{D}$ denotes the deterministic state machine with one state, which is marked, and has self-loops labeled by every event in D. Thus the augmented NSM $\mathcal{P}^D$ can also be obtained by adding self-loops on each state of $\mathcal{P}$ on those events in D that are refused at that state, i.e., $\mathcal{P}^D := (X_\mathcal{P}, \Sigma, \delta_{\mathcal{P}^D}, x^0_\mathcal{P}, X_m)$, where the transition function is defined as:

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

Refer to Example 1 for illustration. Since the trajectory model of $\mathcal{D}$ is $(det(D^*), det(D^*))$, the augmented trajectory model is given by

$$((T^m(\mathcal{P}))^D, (T(\mathcal{P}))^D) := (T^m(\mathcal{P}^D), T(\mathcal{P}^D)) = (T^m(\mathcal{P}) \parallel_\emptyset det(D^*), T(\mathcal{P}) \parallel_\emptyset det(D^*)).$$

It was shown in [30, Proposition 4] and [19, Proposition 3] that whenever the priority sets of a given pair of systems exhaust the entire event set, then the operation of PSC can be reduced to that of SSC of appropriately augmented systems. Specifically, given a pair of trajectory models (P^m, P) and (Q^m, Q) with priority sets $A, B \subseteq \Sigma$, respectively, if $A \cup B = \Sigma$, then

$$P^m \parallel_B Q^m = (P^m)^{\Sigma - A} \parallel_\Sigma (Q^m)^{\Sigma - B}; \quad P \parallel_B Q = P^{\Sigma - A} \parallel_\Sigma Q^{\Sigma - B}.$$

Consequently we have the following identities:

$$L(P^m \parallel_B Q^m) = L((P^m)^{\Sigma - A}) \cap L((Q^m)^{\Sigma - B}); \quad L(P \parallel_B Q) = L(P^{\Sigma - A}) \cap L(Q^{\Sigma - B}).$$

Thus the technique of augmentation is useful in studying the behavior of a pair of systems operating in prioritized synchrony if their priority sets jointly exhaust the entire event set. In particular, we can apply the technique of augmentation in supervisory control, as the event set Σ can be written as the union of the priority set of plant, which is the set of uncontrollable and controllable events, and the priority set of supervisor, which is the set of controllable and driven events.

3 Observation-Compatible Systems

In many control designs, it is not possible to completely observe the behavior of the uncontrolled plant due to lack of sufficient number of sensors. Thus, certain events executed by the uncontrolled plant may be *unobservable*. In the setting of supervisory control, an observation mask—a projection map defined from the set of events to the set of observable events—is used to describe such partial observation. In such situations it is natural to require

that the control actions taken by a supervisor following indistinguishable traces be identical. We call this property of a supervisor *observation-compatibility*. In this section, we formally define the notion of observation-compatibility of the trajectory model of a nondeterministic discrete event system, and study some of its properties.

Let $\Sigma^o \subseteq \Sigma$ be the set of *observable* events, i.e., the events that can be sensed by a supervisor. A projection function $M : \Sigma \to \Sigma^o \cup \{\epsilon\}$, called an *observation mask* [21, 6], is used to represent such a partial observation; it is defined as:

$$\forall \sigma \in \Sigma : M(\sigma) := \begin{cases} \sigma & \text{if } \sigma \in \Sigma^o \\ \epsilon & \text{otherwise} \end{cases}$$

Note that we assume that the observation mask is a projection function.

Recall from [26] that a language $K \subseteq \Sigma^*$ is said to be controllable with respect to a given prefix-closed language H and the set of uncontrollable events $\Sigma - B$, called $(H, \Sigma - B)$-*controllable*, if

$$pr(K)(\Sigma - B) \cap H \subseteq pr(K),$$

i.e., if the extension of a certain prefix of K by an uncontrollable event results in a trace of H, then this extended trace should also be a prefix of K. Also, recall from [21] that K is is said to be observable with respect to H and a given observation mask $M(\cdot)$, called (H, M)-*observable*, if

$$\forall s, t \in pr(K), \sigma \in \Sigma : M(s) = M(t), s\sigma \in pr(K), t\sigma \in H \Rightarrow t\sigma \in pr(K).$$

In other words, K is said to be (H, M)-observable if given an indistinguishable pair of traces in $pr(K)$, the pair of traces resulting from appending a common event to the given pair has identical membership in $pr(K)$ whenever they have identical membership in H. It was shown in [21] that the observability of prefix-closed languages is preserved under intersection so that the *infimal prefix-closed and observable superlanguage* of a given language exists. Using the above notion of observability we next define the concept of observation-compatibility.

Definition 1 Given a trajectory model (S^m, S) and an observation mask $M(\cdot)$, (S^m, S) is said to be *observation compatible with respect to* $M(\cdot)$ or simply M-*compatible* if

$$\forall s, t \in L(S), \sigma \in \Sigma : M(s) = M(t), s\sigma \in L(S) \Rightarrow t\sigma \in L(S).$$

A NSM is said to be M-compatible, if its associated trajectory model is M-compatible.

Thus a trajectory model is M-compatible if and only if its generated language is (Σ^*, M)-observable. Note that the property of *observation-compatibility* captures physically realizable supervisors. Such supervisors make control decisions based on *only* the observed event trace of the system, and do not require any "special" internal knowledge of the system. Next we show that M-compatibility of a *deterministic* trajectory model is preserved under augmentation. We first need to establish two lemmas.

Definition 2 Given a nonempty prefix-closed language K, the *projection of* Σ^* *onto* K is defined inductively by:

$$\pi_K(\epsilon) := \epsilon; \quad \forall s \in \Sigma^*, \sigma \in \Sigma : \pi_K(s\sigma) := \begin{cases} \pi_K(s)\sigma & \text{if } \pi_K(s)\sigma \in K \\ \pi_K(s) & \text{otherwise} \end{cases}$$

When the choice of K is clear, we use the abbreviated notation s' for $\pi_K(s)$.

The first lemma asserts that the state reached by the execution of a certain trace in an augmented deterministic state machine is the same as that reached in the unaugmented state machine by the execution of the trace projected onto its language.

Lemma 1 Let $\mathcal{P} := (X_\mathcal{P}, \Sigma, \delta_\mathcal{P}, x^0_\mathcal{P}, X^m_\mathcal{P})$ be a deterministic state machine and $D \subseteq \Sigma$. Then for each $s \in L(\mathcal{P}^D)$, $\delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, s) = \delta^*_\mathcal{P}(x^0_\mathcal{P}, \pi_{L(\mathcal{P})}(s))$.

Proof: We use induction on length of s for proving the assertion. For notational simplicity, define $\pi_{L(\mathcal{P})}(s) := s'$. If $|s| = 0$, then $s = s' = \epsilon$. Hence $\delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, s) = \delta^*_\mathcal{P}(x^0_\mathcal{P}, s') = x^0_\mathcal{P}$, as $\mathcal{P}$, and so $\mathcal{P}^D$, are deterministic. Thus the base step trivially holds. In order to prove the induction step, suppose $s = \bar{s}\sigma$, where $\sigma \in \Sigma$. Define $\bar{s}' := \pi_{L(\mathcal{P})}(\bar{s})$. Then it follows from induction hypothesis that $\delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, \bar{s}) = \delta^*_\mathcal{P}(x^0_\mathcal{P}, \bar{s}') := x_{\bar{s}}$. If $\sigma \notin \Re_\mathcal{P}(x_{\bar{s}})$, then $\delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, s) = \delta^*_\mathcal{P}(x_{\bar{s}}, \sigma) = \delta^*_\mathcal{P}(x^0_\mathcal{P}, s')$. On the other hand, if $\sigma \in D \cap \Re_\mathcal{P}(x_{\bar{s}})$, then $s' = \bar{s}'$, and $\delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, s) = x_{\bar{s}}$, so that $\delta^*_\mathcal{P}(x^0_\mathcal{P}, s') = \delta^*_\mathcal{P}(x^0_\mathcal{P}, \bar{s}') = x_{\bar{s}} = \delta^*_{\mathcal{P}^D}(x^0_\mathcal{P}, s)$. This proves the induction step and completes the proof. ∎

The next lemma asserts that if a certain language is (Σ^*, M)-observable, then the indistinguishability of a pair of traces implies indistinguishability of their projections onto the language.

Lemma 2 Consider an observation mask $M(\cdot)$, and a nonempty prefix-closed language $K \subseteq \Sigma^*$. If K is (Σ^*, M)-observable, then

$$\forall s, t \in \Sigma^* : M(s) = M(t) \Rightarrow M(\pi_K(s)) = M(\pi_K(t)).$$

Proof: For notational simplicity, define $s' := \pi_K(s)$ and $t' := \pi_K(t)$. We prove the assertion by induction on $|s| + |t|$. For the base step, if $|s| = 0$ or $|t| = 0$, then $M(s) = M(t) = \epsilon$, so $M(s') = M(t') = \epsilon$. For the induction step, consider $s = \bar{s}\sigma_s$ and $t = \bar{t}\sigma_t$ with $\bar{s}, \bar{t} \in \Sigma^*$ and $\sigma_s, \sigma_t \in \Sigma$. Define $\bar{s}' := \pi_K(\bar{s})$ and $\bar{t}' := \pi_K(\bar{t})$. We have three possibilities: (i) $M(\sigma_s) = \epsilon$, which implies that $M(\bar{s}) = M(t)$. Then, $M(s') = M(\bar{s}') = M(t')$, where the first equality follows trivially from the unobservability of σ_s and the second equality follows by induction hypothesis. (ii) $M(\sigma_t) = \epsilon$. Then it follows from symmetry and case (i) above that $M(s') = M(t')$. (iii) $M(\sigma_s) \neq \epsilon$, $M(\sigma_t) \neq \epsilon$, which implies that $\sigma_s = \sigma_t := \sigma$ and $M(\bar{s}) = M(\bar{t})$. By induction hypothesis, $M(\bar{s}') = M(\bar{t}')$. Since K is (Σ^*, M)-observable, either $\bar{s}'\sigma, \bar{t}'\sigma \in K$ or $\bar{s}'\sigma, \bar{t}'\sigma \notin K$. In the first case, $M(s') = M(\bar{s}'\sigma) = M(\bar{s}')\sigma = M(\bar{t}')\sigma = M(\bar{t}'\sigma) = M(t')$. In the second case, $M(s') = M(\bar{s}') = M(\bar{t}') = M(t')$. ∎

The results of Lemma 1 and 2 are now used to prove that the observation-compatibility of a deterministic system is preserved under augmentation.

Theorem 1 Let (S^m, S) be a deterministic trajectory model, $M(\cdot)$ be an observation mask, and $D \subseteq \Sigma$. Suppose (S^m, S) is M-compatible. Then $((S^m)^D, S^D)$ is also M-compatible (and deterministic).

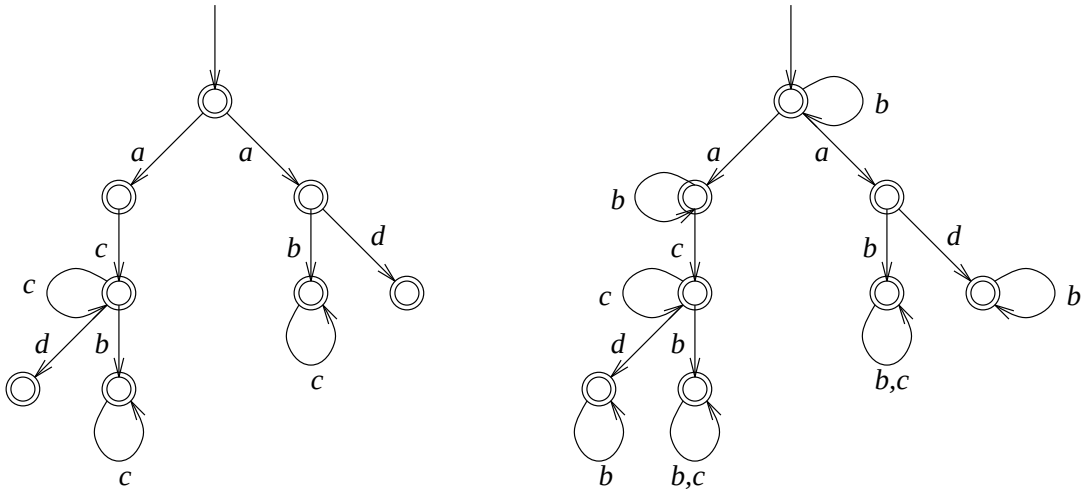
Proof: It suffices to show that $L(S^D)$ is (Σ^*, M)-observable. Pick $s, t \in L(S^D), \sigma \in \Sigma$ such that $M(s) = M(t)$ and $s\sigma \in L(S^D)$. Then we need to show that $t\sigma \in L(S^D)$. Since (S^m, S) is a deterministic trajectory model, there exists a deterministic state machine $\mathcal{S} := (X_{\mathcal{S}}, \Sigma, \delta_{\mathcal{S}}, x^0_{\mathcal{S}}, X^m_{\mathcal{S}})$ with trajectory model (S^m, S). Then $((S^m)^D, S^D) = (T^m(\mathcal{S}^D), T(\mathcal{S}^D))$. Define $s' := \pi_{L(\mathcal{S})}(s)$ and $t' := \pi_{L(\mathcal{S})}(t)$. Then it follows from Lemma 1 that

$$\delta^*_{\mathcal{S}^D}(x^0_{\mathcal{S}}, s) = \delta^*_{\mathcal{S}}(x^0_{\mathcal{S}}, s'); \quad \delta^*_{\mathcal{S}^D}(x^0_{\mathcal{S}}, t) = \delta^*_{\mathcal{S}}(x^0_{\mathcal{S}}, t'). \tag{1}$$

Also, since $M(s) = M(t)$, it follows from Lemma 2 that $M(s') = M(t')$. Hence if $s'\sigma \in L(S)$, then it follows from (Σ^*, M)-observability of $L(S)$ that $t'\sigma \in L(S)$. Hence (1) implies $t\sigma \in L(S^D)$. On the other hand, if $s'\sigma \notin L(S)$, then $\sigma \in D$, so $t\sigma \in L(S^D)$ trivially. ■

We show via the following example that the requirement of determinism cannot be relaxed in Theorem 1.

Example 1 In order to see that the determinism is a necessary condition for Theorem 1



(a) NSM P (b) Augmented NSM $P^{\{b\}}$

Figure 1: Diagram Illustrating Example 1

to hold, consider the NSM $\mathcal{P}$ shown in Figure 1(a) with $\Sigma = \{a, b, c, d\}$ and $M(\cdot)$ such that $M(a) = a, M(b) = b, M(c) = \epsilon, M(d) = d$. Then (i) $ac^* \in L(\mathcal{P})$, and each trace in ac^* has identical mask value. It can be checked that the set of events enabled after each trace in ac^* equals $\{b, c, d\}$. (ii) $ac^*bc^* \in L(\mathcal{P})$, and each trace in ac^*bc^* has identical mask value. It can also be checked that the set of events enabled after each trace in ac^*bc^* equals $\{c\}$. (iii) Finally, $ac^*d \in L(\mathcal{P})$, and each trace in ac^*d has identical mask value. One can verify that

no event is enabled after each such trace. Thus $L(\mathcal{P})$ is (Σ^*, M)-observable, so the associated trajectory model $(T^m(\mathcal{P}), T(\mathcal{P}))$ is M-compatible.

The augmented NSM $\mathcal{P}^{\{b\}}$ is shown in Figure 1(b). Then $ab, abc \in L(\mathcal{P}^{\{b\}})$ with $M(ab) = M(abc)$. However, the set of events enabled after ab equals $\{b, c\}$, whereas the set of events enabled after abc equals $\{b, c, d\}$. Thus $L(\mathcal{P}^{\{b\}})$ is not (Σ^*, M)-observable, and so the associated trajectory model $(T^m(\mathcal{P}^{\{b\}}), T(\mathcal{P}^{\{b\}})$ is not M-compatible. ■

4 Centralized Control under Partial Observation

In a previous paper [30], we showed that PSC can be used as a mechanism of control under the restriction that all controllable events are observable to the supervisor. We show in this section that PSC can be used as a mechanism of control without imposing this restriction on the observation mask. As discussed in the previous section, whenever the observations of a supervisor are filtered through a mask, the supervisor must be observation-compatible with respect to its observation mask, i.e., a supervisor under partial observation must satisfy the constraint that following each pair of traces that look alike under the observation mask, it must take identical control action.

Prior to establishing the main result of this section, we prove the following preliminary result:

Lemma 3 Let $H \subseteq \Sigma^*$ be prefix-closed, $K \subseteq H$, and $M(\cdot)$ be an observation mask. If $K^M \subseteq \Sigma^*$ denotes the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K, then $K^M \cap H$ equals the infimal prefix-closed and (H, M)-observable superlanguage of K.

Proof: For simplicity of notation define $K' := K^M \cap H$. Let $\hat{K} \subseteq \Sigma^*$ denote the infimal prefix-closed and (H, M)-observable superlanguage of K. We need to show that $K' = \hat{K}$. In order to show that $\hat{K} \subseteq K'$, it suffices to show that K' is a prefix-closed (H, M)-observable superlanguage of K. Since $K \subseteq H$, it follows that $K' = K^M \cap H$ is a superlanguage of K. Also, from the fact that prefix-closure is preserved under intersection, it follows that K' is prefix-closed. Finally, since K^M is (Σ^*, M)-observable, clearly, it is (H, M)-observable. Then it follows from the fact that observability of prefix-closed languages is preserved under intersection [21, 28] that $K' = K^M \cap H$ is also (H, M)-observable.

It remains to show that $K' \subseteq \hat{K}$. Suppose for contradiction that $\hat{K}$ is a proper sublanguage of K'. Then there exists $s \in \hat{K}$ and $\sigma \in \Sigma$ such that $s\sigma \in K' - \hat{K}$. Since $K' \subseteq K^M$, it follows that $s\sigma \in K^M$. Also, since $K \subseteq \hat{K} \subset K' \subseteq K^M$, and K^M is the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K, it follows that K^M is also the infimal prefix-closed and (Σ^*, M)-observable superlanguage of $\hat{K}$. Finally, since $s \in \hat{K}$, $s\sigma \notin \hat{K}$, and $s\sigma \in K^M$, it follows from the fact that K^M is the infimal prefix-closed and (Σ^*, M)-observable superlanguage of $\hat{K}$ that there exists $t \in \hat{K}$ such that $M(t) = M(s)$ and $t\sigma \in \hat{K}$. We also have that $s \in \hat{K}$, and $s\sigma \in K' - \hat{K} \subseteq H - \hat{K}$. Thus we arrive at a contradiction to the fact that $\hat{K}$ is (H, M)-observable. ■

The following corollary is immediate from Lemma 3:

Corollary 1 Let $H \subseteq \Sigma^*$ be prefix-closed, $M(\cdot)$ be an observation function, and $K \subseteq H$ be prefix-closed and (H, M)-observable. If $K^M \subseteq \Sigma^*$ denotes the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K, then $K^M \cap H = K$.

Recall from [19] that a supervisor with trajectory model (S^m, S) is said to be *non-marking* if $S^m = S$. In the following theorem we obtain a necessary and sufficient condition for the existence of a non-marking and observation-compatible deterministic supervisor. We need the following result from [30, Remark 11]: Given a plant trajectory model (P^m, P) with priority set A, if a language K satisfies the controllability condition of Theorem 2 below, and H is any prefix-closed language satisfying $L(P^{\Sigma-A}) \cap H = K$, then the non-marking deterministic supervisor $(S, S) := (det(H), det(H))$ with priority set B such that $A \cup B = \Sigma$ yields K as the closed-loop behavior $L(P\ {}_A\|_B\ S)$.

Theorem 2 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$, with $A \cup B = \Sigma$, $M(\cdot)$ be an observation mask, and $K \subseteq L(P^{\Sigma-A})$ be a nonempty language. Then there exists a deterministic, non-marking, and M-compatible supervisor with trajectory model (S, S) such that $L(P\ {}_A\|_B\ S) = K$ if and only if

> **Prefix-closure:** $pr(K) = K$, and
> **Controllability:** $pr(K)(\Sigma - B) \cap L(P^{\Sigma-A}) \subseteq pr(K)$, and
> **Observability:** $\forall s, t \in pr(K), \sigma \in \Sigma : M(s) = M(t), s\sigma \in pr(K), t\sigma \in L(P^{\Sigma-A}) \Rightarrow t\sigma \in pr(K)$.

In this case S can be chosen to be $det(K^M)$, where K^M is the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K.

Proof: In order to see the sufficiency part, consider the supervisor with $S := det(K^M)$. Then $L(S) = K^M$, so that S is M-compatible. Also, it follows from Corollary 1 that $K^M \cap L(P^{\Sigma-A}) = K$. Using [30, Remark 11], we obtain $L(P\ {}_A\|_B\ S) = K$.

In order to see the necessity part, suppose (S, S) is the trajectory model of a deterministic non-marking and M-compatible supervisor such that $L(P\ {}_A\|_B\ S) = K$. Then it follows from the necessity part of [30, Theorem 4] that K is prefix-closed and controllable. It remains to show that K is $(L(P^{\Sigma-A}), M)$-observable. Since $K = L(P\ {}_A\|_B\ S) = L(P^{\Sigma-A}) \cap L(S^{\Sigma-B})$, it suffices to show that $L(S^{\Sigma-B})$ is (Σ^*, M)-observable. This follows from the fact that (S, S) is a deterministic and M-compatible trajectory model, and as shown in Theorem 1, M-compatibility of deterministic trajectory models is preserved under augmentation. ■

Remark 1 In contrast to the standard controllability and observability condition of the Ramadge-Wonham setting, the conditions of Theorem 2 refer to the language of the *augmented* plant. This language depends on the *trajectory model* of the plant and in general cannot be deduced from the language model of the plant. Readers are referred to [30, Remark 9, Example 3] for further elaboration on this point.

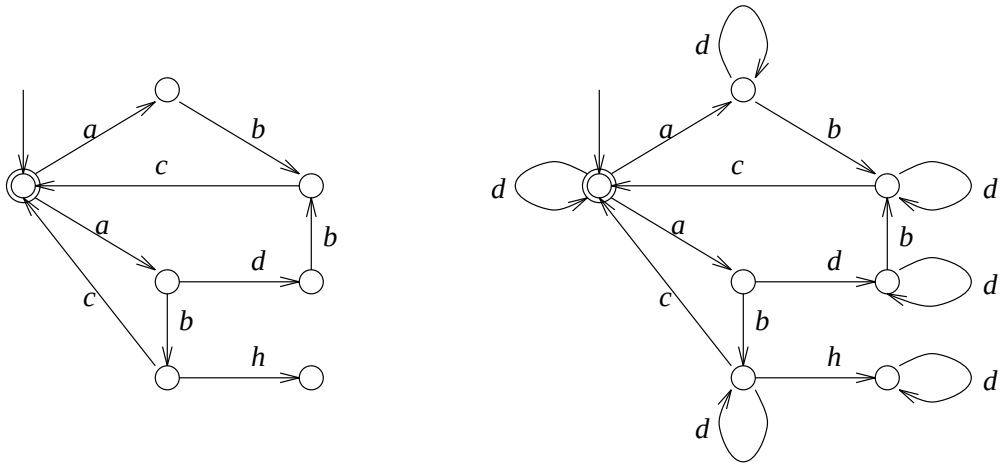
Also, since the necessity part of Theorem 2 uses the result of Theorem 1, it follows from Example 1 that the necessity part of Theorem 2 may not hold if the supervisor is not required

to be deterministic. In a recent paper Inan has studied the design of nondeterministic supervisors under partial observation [13], where he has introduced the notion of *co-closure* (a condition weaker than controllability and observability combined), and has proved its necessity and sufficiency.

Finally, it may seem that the result of Theorem 2 is an immediate consequence of our prior work on nondeterministic systems, and the standard supervisory control results. However, this is not true as it is not clear at the outset whether our results on nondeterministic systems under *complete* observations will immediately "carry over" to the case of *partial* observations (with appropriate extensions as in the standard supervisory control). In fact the result of Theorem 2 fails to hold if more general *non-projection* type observation masks are considered. This is because the observation-compatibility of a deterministic system is not preserved under augmentation if the observation mask is no longer the projection type. To see this consider an observation mask that identifies the only events a and b of a deterministic system which executes the event a in its initial state and deadlocks. Clearly, the system is observation-compatible. However, its augmentation with the event b has a self-loop on b in both its states. So, in the augmented system a as well as b can occur after the occurrence of the initial b, whereas only b can occur after the occurrence of the initial a, which violates the observation-compatibility since a and b are indistinguishable.

We next apply the result of Theorem 2 to the design of a supervisor that achieves mutually exclusive usage of a shared communication channel in a communication system.

Example 2 Consider the nondeterministic plant P depicted in Figure 2(a). In this example,



(a) Plant P (b) Augmented plant $P^{\{d\}}$

Figure 2: Plant P and augmented plant $P^{\Sigma - A} = P^{\{d\}}$

the plant represents a partial model of a multi-user communication system. Only the portions of the model needed to illustrate the main result are included. The communication system

has two channels. The first user can transmit messages using either channel, and switches between the channels in a manner that is unmodeled and hence nondeterministic. The second user can transmit only on channel 2. The event a represents the commencement of transmission by user 1 and results in a nondeterministic transition to one of two successor states depending on which channel is used. The event b represents the commencement of transmission by user 2. Both the commencement events are controllable but are unobservable to the supervisor to be constructed. If both users are able to transmit their messages without collision, then an uncontrollable completion event c occurs which returns the plant to its initial state.

In order to avoid collision of messages, user 1 may receive a signal that causes it to vacate channel 2 provided it has in fact chosen channel 2. This is represented by the event d. It is a driven event because it must be initiated by a supervisor and is executed synchronously by the plant only if able to do so–i.e., only if user 1 is transmitting on channel 2. If user 1 has been transmitting on channel 2 and user 2 commences transmission without it being preceded by d, then there are two possibilities: If user 1 has happened to finish before user 2 starts, then b is followed by the completion event c; otherwise b is followed by the collision event h, an uncontrollable event.

Thus in this example,

$$\Sigma = \{a, b, c, d, h\},\ A = \{a, b, c, h\},\ B = \{a, b, d\},$$

as a and b are controllable, c and h are uncontrollable, and d is a driven event. Note that a and b are the only events that are unobservable to the supervisor to be constructed. The basic performance specification is that a collision-free service should be provided. This can be represented by the prefix-closed sublanguage of the augmented plant (shown in Figure 2(b)) given by

$$K_0 := \{s \in L(P^{\Sigma - A}) \mid s \text{ does not contain } h\} = pr[(d^*ad^*bd^*c)^*].$$

However, since user 1 cannot vacate the channel 2 unless it is using it, it is reasonable to consider the desired behavior to be the sublanguage of K_0 consisting of those traces that do not contain any occurrence of d that is not immediately preceded by a. This is given by

$$K_1 := pr[(abc + adbc)^*].$$

Since the uncontrollable event h can occur following the trace $ab \in K_1$, it is not controllable. The supremal prefix-closed and controllable sublanguage of K_1 is given by

$$K_1^{\uparrow} = pr[(adbc)^*].$$

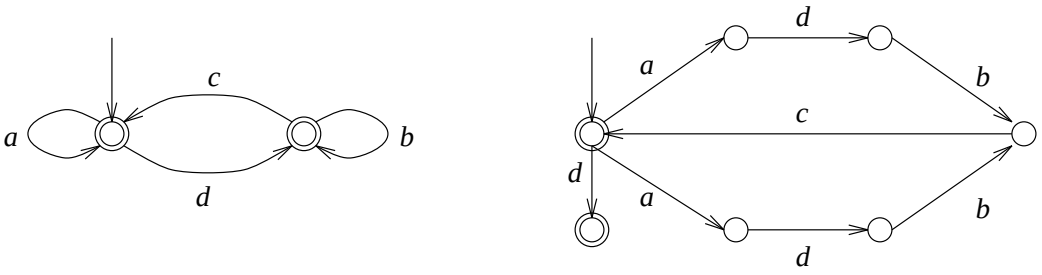
However this is not $L(P^{\Sigma - A}, M)$-observable. In fact since $\epsilon, a \in K_1^{\uparrow}$ with $M(\epsilon) = M(a)$ and $d \in L(P^{\Sigma - A}) - K_1^{\uparrow}$, it follows that any prefix-closed sublanguage of $K_1^{\uparrow}$ that is $(L(P^{\Sigma - A}), M)$-observable cannot contain ad. Thus, a prefix-closed $(L(P^{\Sigma - A}), M)$-observable sublanguage of $K_1^{\uparrow}$ is contained in $pr(a)$. By Theorem 2, it follows any M-compatible supervisor that

results in a closed-loop generated language contained in the specification language K_1 gives a closed-loop generated language contained in $pr(a)$. This is clearly unsatisfactory.

Thus, we must relax the specification given by K_1 keeping in mind that the constraint given by K_0 must be satisfied. The infimal prefix-closed and $(L(P^{\Sigma-A}), M)$-observable superlanguage of $K_1^{\uparrow}$ is $pr[(adbc)^*d]$, which is a sublanguage of K_0. Since $pr[(adbc)^*d]$ is also controllable, and since its infimal prefix-closed and (Σ^*, M)-observable superlanguage is $pr[(a^*db^*c)^*d]$ it follows from Theorem 2 that the non-marking supervisor

$$S_1 := det[(pr(a^*db^*c)^*d)] = det[(pr(a^*db^*c)^*)]$$

depicted in Figure 3(a) is M-compatible and yields $pr[(adbc)^*d]$ as the closed-loop generated language. The closed-loop system is shown in Figure 3(b).



(a) Supervisor 1 (b) Closed-loop system 1

Figure 3: Supervisor S_1 and closed-loop system $P\ {}_A\|_B\ S_1$

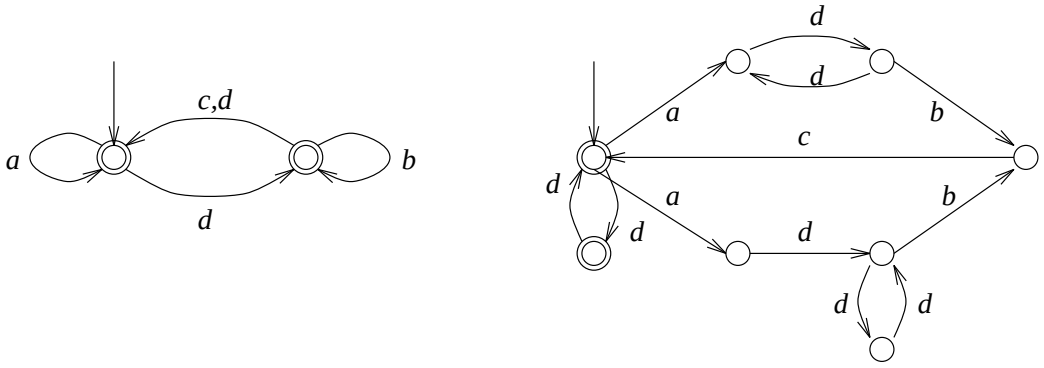
The supervisor implements the following simple control strategy: Initially it allows only user 1 to transmit. Before enabling transmission by user 2, it signals user 1 to vacate channel 2. This command is synchronously executed in the plant only when user 1 is transmitting on channel 2; otherwise, it is "refused" by the plant and occurs asynchronously in the supervisor. The supervisor then allows user 2 to communicate, and returns to its initial state when the completion event c occurs. The ability of the plant to refuse a driven event initiated by the supervisor is essential to our control, and is available because of the PSC-based control design. (Such a feature is certainly unavailable in an SSC-based control design.)

This design is not entirely satisfactory since, as can be seen from Figure 3(b), the closed-loop system deadlocks following the execution of any trace in $(adbc)^*d$. [4] This is because we did not require that the closed-loop behavior be *live* [17]. [5] So the next alternative is to consider a live superlanguage of the "non-live" language $pr[(adbc)^*d]$ that is also controllable and observable and is contained in K_0. Although controllability and observability of prefix-closed languages are preserved under intersection, liveness is not. Similarly, although controllability and liveness of languages is preserved under union, observability is not. Hence, no unique solution can be identified. So a "semi-automatic" design involving some human reasoning is unavoidable.

[4]Note that although the closed-loop system is non-blocking in the sense that the prefix-closure of the recognized refusal-traces is the same as the generated refusal-traces, it may deadlock.

[5]Informally, a language is said to be live if each of its trace has an extension in the language.

With a little insight into the problem, it is easy to see that a simple modification of the supervisor in which a transition is added to permit the supervisor to return to its initial state by execution of d achieves liveness of the closed-loop behavior. The new supervisor, denoted S_2, and the resulting closed-loop system are shown in Figure 4. The closed-loop system can



(a) Supervisor 2 (b) Closed-loop system 2

Figure 4: Supervisor S_2 and closed-loop system $P \ {}_A\|_B \ S_2$

no longer deadlock. The language of the closed-loop system equals $pr[(dd + ad(dd)^*bd^*c)^*]$, which is a sublanguage of K_0 as desired.

Note that both S_1 and S_2 do not change their state when either a or b occur, showing that they are compatible with the unobservability of these events.

We conclude this section by extending the result of Theorem 2 to obtain conditions for the existence of *non-blocking* supervisors. Recall from [19, Definition 6] that given a plant (P^m, P) with priority set A, a supervisor (S^m, S) with priority set B is said to be *language model non-blocking* if $pr(L(P^m \ {}_A\|_B \ S^m)) = L(P \ {}_A\|_B \ S)$; it is said to be *trajectory model non-blocking* if $pr(P^m \ {}_A\|_B \ S^m) = P \ {}_A\|_B \ S$. In the following corollary we provide a necessary and sufficient condition for the existence of an observation-compatible and language model non-blocking supervisor.

Corollary 2 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$, with $A \cup B = \Sigma$, $M(\cdot)$ an observation mask, and $K^m \subseteq L((P^m)^{\Sigma - A})$ a nonempty language. Then there exists a deterministic, non-marking, language model non-blocking, and M-compatible supervisor with trajectory model (S, S) such that $L(P^m \ {}_A\|_B \ S^m) = K^m$ if and only if

> **Relative-closure:** $pr(K^m) \cap L((P^m)^{\Sigma - A}) = K^m$, and
> **Controllability:** $pr(K^m)(\Sigma - B) \cap L(P^{\Sigma - A}) \subseteq pr(K^m)$, and
> **Observability:** $\forall s, t \in pr(K^m), \sigma \in \Sigma : M(s) = M(t), s\sigma \in pr(K^m), t\sigma \in L(P^{\Sigma - A}) \Rightarrow t\sigma \in pr(K^m)$.

In this case S can be chosen to be $det((K^m)^M)$, where $(K^m)^M$ denotes the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K^m.

Proof: First consider sufficiency. Since $pr(K^m)$ is nonempty, prefix-closed, controllable, and $(L(P^{\Sigma-A}), M)$-observable, it follows from the sufficiency part of Theorem 2 that the non-marking supervisor with $S := det((pr(K^m))^M) = det((K^m)^M)$ is M-compatible, and $L(P\ {}_A\|_B\ S) = pr(K^m)$. Hence using the relative closure condition we obtain the following series of equalities:

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

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.

The necessity part follows from the necessity parts of Theorem 2 and [19, Theorem 5]. ■

The result of Corollary 2 can be extended to obtain a necessary and sufficient condition for the existence of an observation-compatible and trajectory model non-blocking supervisor. We need the following result from [19, Proposition 4]: Given a plant (P^m, P) with priority set A and a nonempty language $K^m \subseteq L((P^m)^{\Sigma-A})$, if there exists a deterministic, non-marking and language model non-blocking supervisor (S, S) with priority set B such that $A \cup B = \Sigma$ and $L(P^m\ {}_A\|_B\ S) = K^m$, then

$$P^m\ {}_A\|_B\ det(pr(K^m)) = P^m\ {}_A\|_B\ S; \quad P\ {}_A\|_B\ det(pr(K^m)) = P\ {}_A\|_B\ S.$$

Corollary 3 Let (P^m, P) be the trajectory model of a plant, $A, B \subseteq \Sigma$, with $A \cup B = \Sigma$, $M(\cdot)$ an observation mask, and $K^m \subseteq L((P^m)^{\Sigma-A})$ a nonempty language. Then there exists a deterministic, non-marking, trajectory model non-blocking, and M-compatible supervisor with trajectory model (S, S) such that $L(P^m\ {}_A\|_B\ S^m) = K^m$ if and only if

Relative-closure: $pr(K^m) \cap L((P^m)^{\Sigma-A}) = K^m$, and
Controllability: $pr(K^m)(\Sigma - B) \cap L(P^{\Sigma-A}) \subseteq pr(K^m)$, and
Observability: $\forall s, t \in pr(K^m), \sigma \in \Sigma : M(s) = M(t), s\sigma \in pr(K^m), t\sigma \in L(P^{\Sigma-A}) \Rightarrow t\sigma \in pr(K^m)$, and
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 $det((K^m)^M)$, where $(K^m)^M$ denotes the infimal prefix-closed and (Σ^*, M)-observable superlanguage of K^m.

Proof: The necessity part follows from the necessity part of Corollary 2 and that of [19, Theorem 6]; the sufficiency part follows from the sufficiency part of Corollary 2 and [19, Proposition 4]. ■

5 Decentralized Control

So far we have restricted our attention to the problem of *centralized* control under partial observation. However, in many applications such as manufacturing systems, communication networks, etc., the plant is physically distributed and it is desirable to have decentralized controllers [6, 20, 22, 29, 32], where each controller is able to control a certain set of events and is able to observe certain other events. The problem of decentralized control can be studied quite elegantly in our PSC based approach.

Without any loss of generality we consider the case of "two-decentralization", i.e., given a discrete event plant P with priority set A we consider synthesis of two supervisors S_1 and S_2 with priority sets B_1 and B_2, respectively, which are compatible with their own observation masks $M_1(\cdot)$ and $M_2(\cdot)$, respectively, such that the controlled plant $P \ {}_A\|_{B_1 \cup B_2} \ (S_1 \ {}_{B_1}\|_{B_2} \ S_2)$ satisfies a desired behavior constraint. The priority set of supervisor $S_i (i = 1, 2)$ is B_i, and its observations are filtered through the mask function $M_i(\cdot)$. Thus the events in the set $A \cap B_i$ are the controllable events for S_i, those in the set $A - B_i$ are the uncontrollable events for S_i, and finally those in $B_i - A$ are the driven events for S_i. Also, S_i must be compatible with $M_i(\cdot)$, i.e., its generated language must be (Σ^*, M_i)-observable. Since an event must belong to at least one of the priority sets we have that $A \cup B_1 \cup B_2 = \Sigma$.

For notational simplicity we define $B := B_1 \cup B_2$ and $S := S_1 \ {}_{B_1}\|_{B_2} \ S_2$. Since the events in the set $A - B$ are in the priority set of neither of the supervisors, these represent the uncontrollable events. Thus for decentralized supervision it is expected that the desired behavior be controllable with respect to these uncontrollable events. The remaining events are in the priority set(s) of one or both of the supervisors, however, their enablement/disablement must satisfy the restriction that results from the partial observability of the supervisors. This is captured by the following condition of co-observability, which is similar to the one given by Rudie and Wonham [29]:

Definition 3 Given the priority sets B_1 and B_2 of two supervisors, and their observation masks $M_1(\cdot)$ and $M_2(\cdot)$, respectively, a language $K \subseteq \Sigma^*$ is said to be *co-observable* with respect to a prefix-closed language $H \subseteq \Sigma^*$, called (H, B_1, B_2, M_1, M_2)*–co-observable*, if

$\forall s_1, s_2, t \in pr(K), \sigma \in B_1 \cup B_2$:
(1) $[\sigma \in B_1 - B_2, M_1(s_1) = M_1(t), s_1\sigma \in pr(K), t\sigma \in H] \Rightarrow [t\sigma \in pr(K)]$
(2) $[\sigma \in B_2 - B_1, M_2(s_2) = M_2(t), s_2\sigma \in pr(K), t\sigma \in H] \Rightarrow [t\sigma \in pr(K)]$
(3) $[\sigma \in B_1 \cap B_2, M_1(s_1) = M_1(t), M_2(s_2) = M_2(t), s_1\sigma, s_2\sigma \in pr(K), t\sigma \in H] \Rightarrow$
$[t\sigma \in pr(K)]$

Thus if an event belongs solely to priority set of one of the supervisors and it is enabled following a trace, then it must be enabled following any other trace that is indistinguishable to that supervisor (provided it can occur in the plant). On the other hand, if the event belongs to the common priority set of the supervisors, and it can occur in the plant following a trace which is indistinguishable from a certain trace to the first supervisor, and from another trace to the second supervisor, and the event is enabled following these latter pair of traces,

then the event must also be enabled following the former trace. It is clear that K is co-observable if and only if $pr(K)$ is co-observable. Also, as is the case with observability, co-observability of prefix-closed languages is preserved under intersection [29]; consequently, the infimal prefix-closed and co-observable superlanguage of a given language exists.

We show below that controllability together with co-observability is necessary and sufficient for decentralized supervision. It is clear that observability with respect to each of the masks implies co-observability. Thus a weaker condition than observability with respect to each of the masks is needed for decentralized supervision; this is because the events in the common priority set of the two supervisors can be disabled by either of them. However, if the common priority set is empty, then under the condition of controllability, co-observability is equivalent to observability with respect to each of the masks.

We saw above that the operation of PSC of a pair of systems can be reduced to that of SSC when the priority sets of the two systems exhaust the entire event set. We next prove that this is also the case when more than two systems are involved. We need the following lemma:

Lemma 4 Consider NSM's $\mathcal{S}_1, \mathcal{S}_2$ with priority sets B_1, B_2 respectively. Then

$$(\mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2)^{\Sigma - B} = \mathcal{S}_1^{\Sigma - B_1} \ {}_{\Sigma}\|_{\Sigma} \ \mathcal{S}_2^{\Sigma - B_2},$$

where $B := B_1 \cup B_2$.

Proof: The above lemma follows from the following series of equalities:

$$\begin{aligned}
\mathcal{S}_1^{\Sigma - B_1} \ {}_{\Sigma}\|_{\Sigma} \ \mathcal{S}_2^{\Sigma - B_2} &= (\mathcal{S}_1^{B - B_1})^{\Sigma - B} \ {}_{\Sigma}\|_{\Sigma} \ (\mathcal{S}_2^{B - B_2})^{\Sigma - B} \\
&= [\mathcal{S}_1^{B - B_1} \ {}_{B}\|_{\Sigma - B} \ det((\Sigma - B)^*)] \ {}_{\Sigma}\|_{\Sigma} \ [\mathcal{S}_2^{B - B_2} \ {}_{B}\|_{\Sigma - B} \ det((\Sigma - B)^*)] \\
&= [\mathcal{S}_1^{B - B_1} \ {}_{B}\|_{B} \ \mathcal{S}_2^{B - B_2}] \ {}_{B}\|_{\Sigma - B} \ [det((\Sigma - B)^*) \ {}_{\Sigma - B}\|_{\Sigma - B} \ det((\Sigma - B)^*)] \\
&= [\mathcal{S}_1^{B - B_1} \ {}_{B}\|_{B} \ \mathcal{S}_2^{B - B_2}] \ {}_{B}\|_{\Sigma - B} \ det((\Sigma - B)^*) \\
&= [\mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2]^{\Sigma - B},
\end{aligned}$$

where the first, second, and final equalities follow from definition of augmentation, and the third equality follows from associativity of PSC. ∎

The following corollary is immediate from the above lemma:

Corollary 4 Consider NSM's $\mathcal{P}, \mathcal{S}_1, \mathcal{S}_2$ with priority sets A, B_1, B_2 respectively, such that $A \cup B_1 \cup B_2 = \Sigma$. Then

$$\mathcal{P} \ {}_{A}\|_{B} \ \mathcal{S} = P^{\Sigma - A} \ {}_{\Sigma}\|_{\Sigma} \ [\mathcal{S}_1^{\Sigma - B_1} \ {}_{\Sigma}\|_{\Sigma} \ \mathcal{S}_2^{\Sigma - B_2}],$$

where $B := B_1 \cup B_2$ and $\mathcal{S} := \mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2$.

Proof: Since $A \cup B = \Sigma$, it follows from a PSC property [9, 30, 19] that

$$\mathcal{P} \ {}_{A}\|_{B} \ \mathcal{S} = \mathcal{P}^{\Sigma - A} \ {}_{\Sigma}\|_{\Sigma} \ \mathcal{S}^{\Sigma - B}.$$

Thus the result follows from Lemma 4. ∎

Remark 2 Corollary 4 shows that the operation of PSC of two or more systems can be reduced to that of SSC whenever the priority sets of all the systems jointly exhaust the entire event set. It also follows that under the hypothesis of Corollary 4

$$L(\mathcal{P} \ {}_A\|_B \ \mathcal{S}) = L(\mathcal{P}^{\Sigma - A}) \cap L(\mathcal{S}_1^{\Sigma - B_1}) \cap L(\mathcal{S}_2^{\Sigma - B_2}). \tag{2}$$

Next we establish a relationship between controllability, observability, co-observability and PSC. In the following lemma we prove that if supervisors $\mathcal{S}_1$ and $\mathcal{S}_2$ are M_1-compatible and M_2-compatible, respectively, and both generate $(\Sigma^*, \Sigma - B)$ controllable languages, then the language of $\mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2$ is $(\Sigma^*, \Sigma - B)$-controllable and $(\Sigma^*, B_1, B_2, M_1, M_2)$–co-observable.

Lemma 5 Consider deterministic state machines $\mathcal{S}_1$ and $\mathcal{S}_2$ with priority sets B_1 and B_2 respectively. Suppose $\mathcal{S}_1$ and $\mathcal{S}_2$ are observation-compatible with respect to masks M_1 and M_2 respectively, and $L(\mathcal{S}_1)$ as well as $L(\mathcal{S}_2)$ are $(\Sigma^*, \Sigma - B)$ controllable, where $B := B_1 \cup B_2$. Then $L(\mathcal{S})$, where $\mathcal{S} := \mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2$, is $(\Sigma^*, B_1, B_2, M_1, M_2)$–co-observable and $(\Sigma^*, \Sigma - B)$ controllable.

Proof: In order to see co-observability, pick $s_1, s_2, t \in L(\mathcal{S})$ and $\sigma \in B$, Since $\mathcal{S}_1$ and $\mathcal{S}_2$ are deterministic, $\mathcal{S}$ is also deterministic. Let $(x^1_{s_1}, x^2_{s_1}), (x^1_{s_2}, x^2_{s_2}), (x^1_t, x^2_t)$ denote the states reached in $\mathcal{S}$ after execution of s_1, s_2, t, respectively, where the first coordinate denotes the state reached in $\mathcal{S}_1$ and the second coordinate denotes the state reached in $\mathcal{S}_2$.

In order to prove co-observability of $L(\mathcal{S})$ we must consider the three different cases of the definition of co-observability. First suppose $\sigma \in B_1 - B_2$, $M_1(s_1) = M_1(t)$, and $s_1\sigma \in L(\mathcal{S})$; we need to show that $t\sigma \in L(\mathcal{S})$. Since $\sigma \in B_1 - B_2$ and $s_1\sigma \in L(\mathcal{S})$ it follows that σ is defined at the state $x^1_{s_1}$ of $\mathcal{S}_1$. Then using the result of Lemma 2 and the fact that $\mathcal{S}_1$ is M_1-compatible, we obtain that σ is also defined at the state x^1_t of $\mathcal{S}_1$; which implies that $t\sigma \in L(\mathcal{S})$. It can be argued in a similar manner that the second and third cases of the definition of co-observability also hold.

In order to see controllability, consider $s \in L(\mathcal{S})$ and $\sigma \in \Sigma - B$. Let (x^1_s, x^2_s) be the state reached in $\mathcal{S}$ by execution of s. Then it follows from the controllability of $L(\mathcal{S}_1)$ that σ is defined at state x^1_s of $\mathcal{S}_1$. Hence $s\sigma \in L(\mathcal{S})$. ■

Given a language K, we use K^{BM_i} $(i = 1, 2)$ to denote the infimal prefix-closed, $(\Sigma^*, \Sigma - B)$-controllable and (Σ^*, M_i)-observable superlanguage of K, which exists as the controllability and observability of prefix closed languages is preserved under intersection. The notation $K^{BM_{12}}$ is used to denote the infimal prefix-closed, $(\Sigma^*, \Sigma - B)$-controllable and $(\Sigma^*, M_1, M_2, B_1, B_2)$–co-observable superlanguage of K. The result of Lemma 5 can be used to show that if $\mathcal{S}_1$ generates K^{BM_1}, $\mathcal{S}_2$ generates K^{BM_2}, then $\mathcal{S}$ generates $K^{BM_{12}}$. This we state in the following theorem:

Theorem 3 Let $M_1, M_2, B_1, B_2, K^{BM_1}, K^{BM_2}, K^{BM_{12}}$ be as defined above. Suppose $\mathcal{S}_1$ and $\mathcal{S}_2$ are deterministic state machines with $L(\mathcal{S}_1) = K^{BM_1}$ and $L(\mathcal{S}_2) = K^{BM_2}$. Then $L(\mathcal{S}) = K^{BM_{12}}$, where $\mathcal{S} := \mathcal{S}_1 \ {}_{B_1}\|_{B_2} \ \mathcal{S}_2$ and $B := B_1 \cup B_2$.

Proof: Since $K \subseteq L(\mathcal{S}_1)$ and $K \subseteq L(\mathcal{S}_2)$, it follows that $K \subseteq L(\mathcal{S})$. Also, it follows from Lemma 5 that $L(\mathcal{S})$ is controllable and co-observable. Thus $L(\mathcal{S})$ is a prefix-closed, controllable and co-observable superlanguage of K. Hence we have that $K^{BM_{12}} \subseteq L(\mathcal{S})$. In order to see the reverse containment, it suffices to show that non-zero length strings of $L(\mathcal{S})$ are also in $K^{BM_{12}}$, as the zero length string ϵ does belong to $K^{BM_{12}}$. Thus we need to show that for any string $t \in K^{BM_{12}}$ and an event σ such that $t\sigma \in L(\mathcal{S})$, $t\sigma \in K^{BM_{12}}$. If $\sigma \in \Sigma - B$, then it follows from the prefix-closure and $(\Sigma^*, \Sigma - B)$-controllability of $K^{BM_{12}}$ that $t\sigma \in K^{BM_{12}}$. On the other hand, if $\sigma \in B$, then we show that the following holds:

(1) $[\sigma \in B_1 - B_2] \Rightarrow [\exists s_1 : M_1(s_1) = M_1(t), s_1\sigma \in pr(K)]$
(2) $[\sigma \in B_2 - B_1] \Rightarrow [\exists s_2 : M_2(s_2) = M_2(t), s_2\sigma \in pr(K)]$
(3) $[\sigma \in B_1 \cap B_2] \Rightarrow [\exists s_1, s_2 : M_1(s_1) = M_1(t), M_2(s_2) = M_2(t), s_1\sigma, s_2\sigma \in pr(K)]$,

as this together with $(\Sigma^*, B_1, B_2, M_1, M_2)$–co-observability of $K^{BM_{12}}$ clearly implies that $t\sigma \in K^{BM_{12}}$.

We prove this using induction on length of t. We only prove that the case (1) holds, as the proof for the other two cases is similar. In order to see the base step, set $t = \epsilon$ (note that we do have $\epsilon \in K^{BM_{12}}$) and pick $\sigma \in B_1 - B_2$. Since $t\sigma = \sigma \in L(\mathcal{S})$ and $\sigma \in B_1 - B_2$, it follows from construction of $\mathcal{S}$ that $\sigma \in L(\mathcal{S}_1) = K^{BM_1}$. Since K^{BM_1} is the infimal prefix closed, $(\Sigma^*, \Sigma - B)$-controllable and (Σ^*, M_1)-observable superlanguage of K, and σ is not an uncontrollable event, it follows that there exists a string s_1 such that $s_1\sigma \in pr(K)$ and $M_1(s_1) = M_1(t) = \epsilon$. The other two cases of the base step can be proved analogously.

In order to see the induction step set $t = \bar{t}\bar{\sigma}$ and pick $\sigma \in B_1 - B_2$. Suppose $\bar{\sigma} \in B_1 - B_2$. Then it follows from induction hypothesis that there exists $\bar{s}_1$ such that $s_1' := \bar{s}_1\bar{\sigma} \in pr(K)$ and $M_1(\bar{s}_1) = M_1(\bar{t})$. Let $(x_t^1, x_t^2), (x_{s_1'}^1, x_{s_1'}^2)$ denote the states reached in $\mathcal{S}$ after execution of t, s_1', respectively, where the first coordinate denotes the state reached in $\mathcal{S}_1$ and the second coordinate denotes the state reached in $\mathcal{S}_2$. Since $\sigma \in B_1 - B_2$, we have that σ is defined at state x_t^1. Hence it follows from Lemma 2 and M_1-compatibility of $\mathcal{S}_1$ that σ is also defined at state $x_{s_1'}^1$, which implies that $s_1'\sigma \in L(\mathcal{S})$. Since $s_1' \in pr(K) \subseteq L(\mathcal{S}_1)$ and $\sigma \in B_1 - B_2$, we must have $s_1'\sigma \in L(\mathcal{S}_1) = K^{BM_1}$. Since K^{BM_1} is the infimal prefix-closed controllable and observable superlanguage of K, and σ is not an uncontrollable event, this implies that there exists s_1 such that $s_1\sigma \in pr(K)$ and $M_1(s_1) = M_1(s_1')$. Thus s_1 is the desired string, as $M_1(s_1) = M_1(s_1') = M_1(t)$. ∎

Using the results derived in this section, we are now ready to present a necessary and sufficient condition for decentralized supervision.

Theorem 4 Consider A, B_1, B_2, M_1, M_2 as defined above with $A \cup B_1 \cup B_2 = \Sigma$. Let (P^m, P) be the trajectory model of a plant, and $K \subseteq L(P^{\Sigma - A})$ be a nonempty language. Then there exist deterministic, non-marking, and M_1-compatible supervisor with trajectory model (S_1, S_1) and M_2-compatible supervisor with trajectory model (S_2, S_2) such that $L(P \,{}_A\|_B\, S) = K$, where $S := S_1 \,{}_{B_1}\|_{B_2}\, S_2$ and $B := B_1 \cup B_2$ if and only if

Prefix-closure: $pr(K) = K$, and
Controllability: $pr(K)(\Sigma - B) \cap L(P^{\Sigma - A}) \subseteq pr(K)$, and
Co-observability: K is $(L(P^{\Sigma - A}), B_1, B_2, M_1, M_2)$–co-observable.

In this case S_i $(i = 1, 2)$ can be chosen to be $det(K^{BM_i})$, where K^{BM_i} is the infimal prefix-closed, $(\Sigma^*, \Sigma - B)$-controllable and (Σ^*, M_i)-observable superlanguage of K.

Proof: We begin by proving the necessity. Prefix-closure and controllability conditions follow from the necessity part of [30, Theorem 4]. We need to show that the co-observability condition also holds. It follows from hypothesis and Corollary 4 that $K = L(P \ {}_A\|_B \ S) = L(P^{\Sigma-A}) \cap L(S_1^{\Sigma-B_1}) \cap L(S_2^{\Sigma-B_2})$. Hence it suffices to show that $H := L(S_1^{\Sigma-B_1}) \cap L(S_2^{\Sigma-B_2})$ is $(\Sigma^*, B_1, B_2, M_1, M_2)$–co-observable. Pick $s_1, s_2, t \in H$ and $\sigma \in \Sigma$. We must consider the three different cases of the definition of co-observability. We only consider the first case, as the other cases can be proved in a similar manner. Suppose $\sigma \in B_1 - B_2$, $s_1\sigma \in H$ and $M_1(s_1) = M_1(t)$. We need to show that $t\sigma \in H$. Since $t \in L(S_2^{\Sigma-B_2})$ and $\sigma \in B_1 - B_2 \subseteq \Sigma - B_2$, $t\sigma \in L(S_2^{\Sigma-B_2})$ trivially. It remains to show that $t\sigma \in L(S_1^{\Sigma-B_1})$. This follows from the fact that $S_1^{\Sigma-B_1}$ is M_1-compatible (as S_1 is M_1-compatible and deterministic, and observation-compatibility of deterministic systems is preserved under augmentation). This completes the proof of the necessity part.

In order to see the sufficiency part select $S_1 := det(K^{BM_1})$ and $S_2 := det(K^{BM_2})$. Then S_1 and S_2 are deterministic, S_1 is M_1-compatible and S_2 is M_2-compatible. It remains to show that the controlled plant language equals K. From Theorem 3 we have that $L(S) = K^{BM_{12}}$, where $K^{BM_{12}}$ is the infimal prefix-closed, $(\Sigma^*, \Sigma - B)$-controllable and $(\Sigma^*, B_1, B_2, M_1, M_2)$–co-observable superlanguage of K. Using arguments similar to those in Lemma 3 we can readily conclude that $L(P^{\Sigma-A}) \cap L(S)$ is the infimal prefix-closed, $(L(P^{\Sigma-A}), \Sigma - B)$-controllable and $(L(P^{\Sigma-A}), B_1, B_2, M_1, M_2)$–co-observable superlanguage of K. Hence it follows from the prefix-closure, controllability and co-observability conditions that

$$L(P^{\Sigma-A}) \cap L(S) = K. \tag{3}$$

We need to show that we also have the following equality: $H := L(P^{\Sigma-A}) \cap L(S^{\Sigma-B}) = K$. This follows from (3) and the fact that K is controllable as is shown next.

Since $L(S) \subseteq L(S^{\Sigma-B})$, clearly, $K \subseteq H$. Suppose for contradiction that there exists a string s such that $s \in H - K$. Let s be a minimal-length such string. Since $\epsilon \in K$, we have $s \neq \epsilon$, which implies $s = \bar{s}\sigma$, where $\bar{s} \in K$ and $\sigma \in \Sigma$. Since $\bar{s} \in K$ and $\bar{s}\sigma \notin K$, it must be the case that $\sigma \in \Sigma - B$. This is contradictory to the fact that K is controllable, as we have $\bar{s} \in K$, $\sigma \in \Sigma - B$, $\bar{s}\sigma \in H$, which implies $\bar{s}\sigma \in L(P^{\Sigma-A})$; however, $\bar{s}\sigma \notin K$. This completes the proof. ■

Remark 3 Note that the conditions of controllability and co-observability in Theorem 4 are with regard to the language of the *augmented* plant, which depends on the trajectory model of the plant and cannot be inferred from the language model of the plant. Also, as is the case of the necessity part of Theorem 2, the necessity part of Theorem 4 may not hold if the supervisors are nondeterministic.

Finally, the result of Theorem 4 can be easily extended to obtain conditions for either language model non-blocking or trajectory model non-blocking supervisors. In fact arguments similar to those given in Corollaries 2 and 3 can be used to show that language model

nonblocking supervision would require the condition of relative-closure instead of that of prefix-closure, and a trajectory model nonblocking supervision would require the additional trajectory-closure condition.

6 Conclusion

In this paper we have extended our earlier work on supervisory control of nondeterministic systems using prioritized synchronization as the mechanism of control and trajectory model as the modeling formalism to control under partial observation. The notion of observation-compatibility of trajectory models has been introduced, and necessary and sufficient conditions for the existence of observation-compatible supervisors have been obtained for centralized as well as decentralized supervision. Although these conditions are similar to the standard conditions of controllability, observability, and co-observability found in literature, they are different, as they depend on the trajectory model as opposed to the language model of the plant. Also, our work demonstrates the suitability of PSC based supervisor design for nondeterministic systems under centralized as well as decentralized setting. These results have been applied for the design of a supervisor that achieves a mutually exclusive usage of a communication channel in a communication system.

References

[1] J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Ready-trace semantics for concrete process algebra with the priority operator. *The Computer Journal*, 30(6):498–506, 1987.

[2] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, Cambridge, 1990.

[3] S. Balemi. Input/output discrete event processes and communication delays. *Discrete Event Dynamical Systems: Theory and Applications*, 4(1):41–85, 1994.

[4] 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.

[5] E. Chen and S. Lafortune. Dealing with blocking in supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 36(6):724–735, 1991.

[6] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. Supervisory control of discrete event processes with partial observation. *IEEE Transactions on Automatic Control*, 33(3):249–260, 1988.

[7] 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.

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

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

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

[11] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.

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

[13] K. Inan. Nondeterministic supervision under partial observations. In Guy Cohen and Jean-Pierre Quadrat, editors, *Lecture Notes in Control and Information Sciences 199*, pages 39–48. Springer-Verlag, New York, 1994.

[14] K. Inan and P. Varaiya. Finitely recursive process models for discrete event systems. *IEEE Transactions on Automatic Control*, 33(7):626–639, 1988.

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

[16] 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.

[17] R. Kumar, V. K. Garg, and S. I. Marcus. On supervisory control of sequential behaviors. *IEEE Transactions on Automatic Control*, 37(12):1978–1985, December 1992.

[18] R. Kumar and M. A. Shayman. Supervisory control of nondeterministic systems under partial observation. In *Proceedings of 1994 IEEE Conference on Decision and Control*, Orlando, FL, December 1994. 3649-3654.

[19] R. Kumar and M. A. Shayman. Nonblocking supervisory control of nondeterministic systems via prioritized synchronization. *IEEE Transactions on Automatic Control*, 41(8):1160–1175, August 1996.

[20] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete event systems. *Information Sciences*, 44:199–224, 1988.

[21] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988.

[22] F. Lin and W. M. Wonham. Decentralized control and coordination of discrete-event systems with partial observation. *IEEE Transactions of Automatic Control*, 35(12):1330–1337, December 1990.

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

[24] R. Milner. *Communication and Concurrency*. Prentice Hall, New York, 1989.

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

[26] 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.

[27] 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.

[28] K. Rudie and W. M. Wonham. The infimal prefix closed and observable superlanguage of a given language. *Systems and Control Letters*, 15(5):361–371, 1990.

[29] K. Rudie and W. M. Wonham. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.

[30] 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.

[31] R. J. van Glabbeek. *Comparative Concurrency Semantics, With Refinement of Actions*. PhD thesis, Free University of Amsterdam, 1990.

[32] Y. Willner and M. Heymann. Supervisory control of concurrent discrete-event systems. *International Journal of Control*, 54(5):1143–1169, 1991.