Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Bisimilarity Control under Partial Observation of Deterministic Discrete Event Systems

Changyan Zhou and Ratnesh Kumar (czhou,rkumar@iastate.edu)
Department of Electrical & Computer Engineering
Iowa State University, Ames, IA 50014

Abstract— **The complexity of the general bisimilarity control problem under partial observation is doubly exponential in the product of the plant and the specification sizes [7]. In order to identify a special case where the complexity may be more manageable, we restrict attention to the class of deterministic plants. In this case, the complexity of verifying existence of a controller turns out to be polynomial, whereas that of performing its synthesis is singly exponential. We establish state-controllability (SC) together with state-recognizability (SR) as a necessary and sufficient condition for the existence of a control. The notion of SC was introduced in [8] as an existence condition for the same problem under the restriction of complete observability of events; and it generalizes the notion of language-controllability (LC) from the setting of language-control to bisimilarity-control. In the presence of partial observation, a supervisor is required to be observation-compatible (also called M-compatible), and the additional condition of SR is needed for the existence of such a supervisor. The property of SR is same as bisimilarity with such a system that can be transformed by state-mergers to a M-compatible system, without altering the bisimilarity of the control it exercises. SR generalizes the notion of language-recognizability [1] in a similar manner as SC generalizes LC. We show that SR is polynomially verifiable, and also present an exponential complexity algorithm for synthesizing of a bisimilarity enforcing supervisor.**

Keywords: Discrete event systems, supervisory control, nondeterministic specification, bisimulation equivalence, controllability, partial observation

I. INTRODUCTION

Control of discrete event systems (DESs) for achieving bisimilarity with a specification has been studied recently (see for example, [2], [3], [8], [9], [4] and references therein). In [8], [9] we studied the bisimulation equivalence control problem allowing both the plant as well as the specification to be nondeterministic, while requiring a complete observation of events. We established a small model result showing that a supervisor exists if and only if it exists over a certain finite state space, namely the power set of the Cartesian product of the plant and the specification states. This also showed that the upper bound for computing a bisimilarity enforcing supervisor (when one exists) is double exponential in the product of the plant and the specification states. It turns out that the above results continue to hold even under partial observation of events (see [7]).

The research was supported in part by the National Science Foundation under the grants NSF-ECS-9709796, NSF-ECS-0099851, NSF-ECS-0218207, NSF-ECS-0244732, NSF-EPNES-0323379, and NSF-0424048, and a DoD-EPSCoR grant through the Office of Naval Research under the grant N000140110621.

While the above mentioned small model result establishes the solvability of the bisimilarity control problem, its high computational complexity serves as a motivation for seeking such specializations which possess more practical complexity bounds. Thus far we have identified two different specializations, both assuming a complete observability of events. In [6] we showed that if instead of the bisimilarity one seeks only the similarity, the control problem (both existence and synthesis) becomes polynomially solvable. Similarly, it was shown in [8], [9] that the bisimilarity control problem (both existence and synthesis) remains polynomially solvable when the plant model is deterministic.

Inspired by the result of [8], [9], the present paper investigates the extension of the bisimilarity control of deterministic plants to the setting of partial observations. The goal remains the same as before namely to determine whether the complexity of the control problem remains within manageable bounds. As we show below, the existence condition remains polynomially bounded, whereas the complexity of synthesizing a supervisor is now exponentially bounded. This is still better than the double exponential complexity of the general case, and is comparable to the complexity of language-control under partial observation using deterministic supervisors [5].

It turns out that the presence of partial observation poses a new challenge. Under partial observation, certain events become indistinguishable to a supervisor. As a result the state-updates on indistinguishable events, that are enabled at a common state of a supervisor, have to be identical. That is the supervisor must also be observation compatible (also called M-compatible) besides being Σ_u-compatible. A main contribution of the paper is the identification of the property that the specification must satisfy for the existence of a M-compatible supervisor. We show that exists a bisimilarity enforcing supervisor for a deterministic plant if and only if the specification is simulated by the plant, is state-controllable and state-recognizable. The new condition of state-recognizability introduced in this paper is needed due to the presence of partial observation. We provide a polynomial algorithm for verifying state-recognizability. While the existence of a bisimilarity enforcing supervisor can be determined polynomially in both plant and specification states for deterministic plants, the upper bound for the synthesis we provide is exponential. For space consideration, all proof are omitted.

II. Notation and Preliminaries

In this paper nondeterministic state machines (NSMs) are used to model discrete event systems. A NSM G is a five tuple: $G := (X, \Sigma, \alpha, X_0, X_m)$, where X is its set of states, Σ is its set of events, $\alpha : X \times (\Sigma \cup \{\epsilon\}) \to 2^X$ is its transition function, $X_0 \subseteq X$ is its set of initial states, and $X_m \subseteq X$ is its set of marked states. G is called deterministic if $|X_0| = 1$, $|\alpha(x, \sigma)| \leq 1$ for all $x \in X$ and $\sigma \in \Sigma$, and $|\alpha(x, \epsilon)| = 0$ for all $x \in X$. The size of a state machine, denoted $|G|$, is the total number of states and transitions. For a NSM, $|G| = O(|X|) + O(|X|^2) = O(|X|^2)$, whereas for a deterministic G, $|G| = O(|X|) + O(|X|) = O(|X|)$. For an event set Σ, we use $\overline{\Sigma}$ to denote $\Sigma \cup \{\epsilon\}$. A triple $(x, \sigma, x') \in X \times \overline{\Sigma} \times X$ is called a *transition* if $x' \in \alpha(x, \sigma)$; if $\sigma = \epsilon$, the transition is called an ϵ-transition. Σ^* denotes the set of all finite-length sequences of events from Σ, called *traces* including the trace of zero length, denoted ϵ. The ϵ-closure of $x \in X$, denoted as $\epsilon^*(x)$, is the set of states reached by the execution of zero or more ϵ-transitions from state x. By using ϵ-closure map, we can extend the definition of transition function from events to traces, $\alpha^* : X \times \Sigma^* \to 2^X$, which is defined inductively as:

$$\forall x \in X, \alpha^*(x, \epsilon) := \epsilon^*(x);$$
$$\forall s \in \Sigma^*, \sigma \in \Sigma : \alpha^*(x, s\sigma) := \epsilon^*(\alpha(\alpha^*(x, s), \sigma)),$$

where for $\hat{X} \subseteq X$ and $\hat{\Sigma} \subseteq \Sigma$, $\alpha(\hat{X}, \hat{\Sigma}) := \cup_{x \in \hat{X}, \sigma \in \hat{\Sigma}} \alpha(x, \sigma)$, and $\epsilon^*(\hat{X}) := \cup_{x \in \hat{X}} \epsilon^*(x)$. The language generated (resp., marked) by G, is denoted as $L(G)$ (resp., $L_m(G)$). $L(G)$ is the sequence of events generated starting from the initial state, i.e., $L(G) = \{s \in \Sigma^* \mid \alpha^*(X_0, s) \neq \emptyset\}$, and $L_m(G)$ is the set of generated sequences that end in a marked state, i.e., $L_m(G) = \{s \in L(G) \mid \alpha^*(X_0, s) \cap X_m \neq \emptyset\}$. For $x \in X$, we define $\Sigma(x) := \{\sigma \in \overline{\Sigma} \mid \alpha(x, \sigma) \neq \emptyset\}$ to denote the set of all labels on which transitions are defined at state x.

One way to model control interaction between plant and supervisor is via the synchronous composition of their state machine (or automaton) representations. The *synchronous composition* of two automata G_1 and G_2, where $G_i = (X_i, \Sigma, \alpha_i, X_{0i}, X_{mi})$, is the automaton

$$G_1 \| G_2 = (X_1 \times X_2, \Sigma, \alpha_\|, X_{01} \times X_{02}, X_{m1} \times X_{m2}),$$

where for $x_1 \in X_1, x_2 \in X_2, \sigma \in \overline{\Sigma}$,

$$\alpha_\|((x_1, x_2), \sigma) = \begin{cases} \alpha_1(x_1, \sigma) \times \alpha_2(x_2, \sigma) & \text{if } \sigma \neq \epsilon \\ (\alpha_1(x_1, \epsilon) \times \{x_2\}) \cup (\{x_1\} \times \alpha_2(x_2, \epsilon)) & \text{if } \sigma = \epsilon \end{cases}$$

We also define the *union* of G_1 and G_2 as the automaton

$$G_1 \cup G_2 = (X_1 \cup X_2, \Sigma, \alpha_\cup, X_{01} \cup X_{02}, X_{m1} \cup X_{m2}),$$

where for $x \in X_1 \cup X_2$, and $\sigma \in \overline{\Sigma}$,

$$\alpha_\cup(x, \sigma) := \begin{cases} \alpha_1(x, \sigma) \cup \alpha_2(x, \sigma) & \text{if } x \in X_1 \cap X_2 \\ \alpha_1(x, \sigma) & \text{if } x \in X_1 - X_2 \\ \alpha_2(x, \sigma) & \text{if } x \in X_2 - X_1 \end{cases}$$

The events executed by a plant are partially observed by a supervisor owing to the type of event-sensors used. Such a partial observation is represented using an observation mask function $M : \overline{\Sigma} \to \overline{\Delta}$ (Δ is the set of observed symbols), satisfying $M(\epsilon) = \epsilon$. M is said to be projection type if $\Delta \subseteq \Sigma$. $\sigma \in \Sigma$ is said to be an unobservable event if $M(\sigma) = \epsilon$, and otherwise it is said to be an observable event. Two events $\sigma_1, \sigma_2 \in \overline{\Sigma}$ are said to be indistinguishable if $M(\sigma_1) = M(\sigma_2)$. The observation mask M is extended to be defined over traces in Σ^* as follows: $M(\epsilon) := \epsilon; \quad \forall s \in \Sigma^*, \sigma \in \Sigma : M(s\sigma) := M(s)M(\sigma)$.

Bisimulation equivalence is a type of behavioral equivalence that is used to describe equivalence between nondeterministic systems. A bisimulation relation over two state machines is a symmetric simulation relation. We next introduce the notion of a simulation relation.

Definition 1: Given automata $G_1 = (X_1, \Sigma, \alpha_1, X_{01}, X_{m1})$ and $G_2 = (X_2, \Sigma, \alpha_2, X_{02}, X_{m2})$, a *simulation relation* is a binary relation $\Phi \subseteq (X_1 \cup X_2)^2$ such that for $x_1, x_2 \in X_1 \cup X_2$, $(x_1, x_2) \in \Phi$ implies

1) $\sigma \in \overline{\Sigma}, x_1' \in \alpha_\cup^*(x_1, \sigma)$
   $\Rightarrow \exists x_2' \in \alpha_\cup^*(x_2, \sigma)$ such that $(x_1', x_2') \in \Phi$.
2) $x_1 \in X_{m1} \cup X_{m2} \Rightarrow x_2 \in X_{m1} \cup X_{m2}$.

G_1 is said to be simulated by G_2, denoted as $G_1 \sqsubseteq_\Phi G_2$, if there exists a simulation relation $\Phi \subseteq (X_1 \cup X_2)^2$ such that for all $x_{01} \in X_{01}$, exists $x_{02} \in X_{02}$ with $(x_{01}, x_{02}) \in \Phi$. This last fact is concisely written as $X_{01} \sqsubseteq_\Phi X_{02}$.
We write $x_1 \sqsubseteq_\Phi x_2$ to denote that there exists a simulation relation Φ with $(x_1, x_2) \in \Phi$, read as x_1 is simulated by x_2. We sometimes omit the subscript Φ from $\sqsubseteq_\Phi$ when it is clear from the context. Further, a simulation relation is called a *bisimulation* equivalence relation if it is symmetric. For a bisimulation equivalence relation Φ if $(x_1, x_2) \in \Phi$, then x_1 and x_2 are called *bisimilar*, written as $x_1 \simeq_\Phi x_2$ (or simply $x_1 \simeq x_2$ when Φ is clear from context). Two automata G_1 and G_2 are said to be bisimilar, denoted as $G_1 \simeq_\Phi G_2$, if exists a bisimulation relation such that $X_{01} \simeq_\Phi X_{02}$.

III. Relative M-Compatibility and State-Recognizability

In our earlier work on bisimilarity control under complete observation [9], we showed that when plant is deterministic the existence as well as synthesis of supervisor can be performed polynomially. We show that even under partial observation, the existence of supervisor remains polynomially verifiable. We use $G = (X, \Sigma, \alpha, X_0, X_m)$, $R = (Q, \Sigma, \delta, Q_0, Q_m)$, and $S = (Y, \Sigma, \beta, Y_0, Y_m)$ to denote the (nondeterministic) plant, specification, and supervisor, respectively. The controlled system is denoted by $G \| S = (X \times Y, \Sigma, \gamma, X_0 \times Y_0, X_m \times Y_m)$.

To motivate the case of deterministic plant, we introduce the following manufacturing example, a solution of which is discussed later.

Example 1: A robot is available at its home location to traverse on one of the two available rails (Figure 1). Traversal on rail-i ($i = 1, 2$) is denoted by event a_i, and while on rail-i, the robot can pick a part from storage-i (event b_i) or storage-$(i+1)$ (event b_{i+1}). The robot then takes the part to work location (event c). Upon completion of processing,

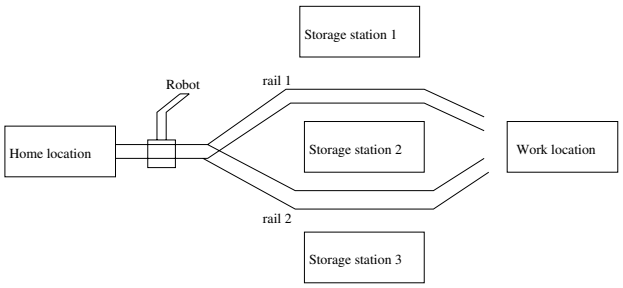

Fig. 1. A manufacturing system

the robot returns the part to either storage-i (event b_i) or storage-$(i+1)$ (event b_{i+1}) and returns to its home location (event d). Not returning the part to its original pick-up location is undesirable. To avoid this undesirable behavior, the specification requires that while the robot is in its home location it be (nondeterministically) decided whether to use storage-i or storage-$(i+1)$ while traversing on rail-i. Plant model G and specification model R are given in Figure 3.

We assume that all events are controllable. However, only events c and d are completely observable. Events a_1 and a_2 are observationally indistinguishable. Also, events b_1, b_2 and b_3 are also observationally indistinguishable. I.e., $\Sigma_u = \emptyset$, $M(a_1) = M(a_2)$ and $M(b_1) = M(b_2) = M(b_3)$. Our goal is to obtain a (Σ_u, M)-compatible supervisor S such that $G \| S \simeq R$.

The notion of (Σ_u, M)-compatibility is formally defined as follows.

Definition 2: Let $\Sigma_u \subseteq \Sigma$ be the set of uncontrollable events and $M : \overline{\Sigma} \to \overline{\Delta}$ be the observation mask, then

- S is called Σ_u-*compatible* if $\forall y \in Y$ and $\forall a \in \Sigma_u$, $\beta(y, a) \neq \emptyset$.
- S is called M-*compatible* if $\forall y \in Y$ and $\forall a, b \in \Sigma(y)$, if $M(a) = M(b)$, then $\beta(y, a) = \beta(y, b)$, where it is assumed that a silent transition is implicitly defined as a self-loop.
- S is called (Σ_u, M)-*compatible* if S is Σ_u-compatible and M-compatible.

Recall from [9] that when G is deterministic, exists a Σ_u-compatible S such that $G \| S \simeq R$ if and only if R is simulated by G (written, "R is G-simulated" for short), and R is state-controllable (SC). State-controllability is a generalization of language-controllability from the setting of deterministic specifications to nondeterministic ones, and is defined as follows:

Definition 3: Given plant G and specification R, we say R is *state-controllable (SC) with respect to G and Σ_u* if

$$\begin{aligned} & s \in L(R), \sigma \in \Sigma_u \text{ such that } s\sigma \in L(G) \\ \Rightarrow \quad & \forall q \in \delta^*(Q_0, s), \sigma \in \Sigma(q). \end{aligned}$$

In the presence of partial observation, S must also be M-compatible. The question becomes what additional property must R possess when there exists (Σ_u, M)-compatible S such that $G \| S \simeq R$. Since R is bisimilar to $G \| S$, we first ask what property $G \| S$ satisfies given that G is deterministic and S is M-compatible. Recall that M-compatibility of S implies indistinguishable events when defined at a state must have identical successors. Clearly, when S is composed with G such a property may no longer hold. However, the composed system $G \| S$ remains "M-compatible relative to G", i.e., if we merge the non-identical successors of indistinguishable events defined at a state of $G \| S$ in a certain way, we can construct a M-compatible state machine that also enforces R when used as a supervisor for G. With this motivation we introduce the notion of relative M-compatibility capturing the property $G \| S$ satisfies whenever G is deterministic and S is M-compatible.

It turns out that relative M-compatibility is not preserved under bisimilarity. So even when it holds that $R \simeq G \| S$, R may not be relative M-compatible. But it is certainly bisimilar to the relative M-compatible system $G \| S$. We name the property of "relative M-compatible—bisimilar" to be state-recognizability (SR) (for reasons to be explained below). The property of SR turns out to be the desired additional property of R required for the existence of a bisimilarity enforcing (Σ_u, M)-compatible supervisor for a deterministic plant.

Next we formally define relative M-compatibility such that it is possible to transform a relative M-compatible system into a M-compatible one with no effect on the control exercised, and show that when S is M-compatible and G is deterministic, $G \| S$ is relative M-compatible. Relative M-compatibility is defined in terms of a "relative M-compatible bisimulation relation". The idea behind defining such a relation is that the relative M-compatible bisimilar state-pairs can be combined to ensure M-compatibility (without effecting the control exercised). Let $X_{syn}(y) := \{x \in X \mid \exists s \in \Sigma^*, x \in \alpha^*(X_0, s) \text{ and } y \in \beta^*(Y_0, s)\}$ represent the set of states in G that synchronize (or are reached by same trace s) with a state y in S.

Definition 4: Given NSMs $G = (X, \Sigma, \alpha, X_0, X_m)$ and $S = (Y, \Sigma, \beta, Y_0, Y_m)$, a symmetric relation Φ over states of S is said to be a *relative M-compatible bisimulation relation* if $(y_1, y_2) \in \Phi$ implies

1) $x_i \in X_{syn}(y_i)$ implies $\Sigma(x_i) \cap \Sigma(y_j) \subseteq \Sigma(x_i) \cap \Sigma(y_i)$ for $i, j = 1, 2$,
   (If x_i can synchronize with y_i in $G \| S$ and is made to synchronize with y_j $(i, j = 1, 2)$, then events enabled at x_i should not change.)
2) $a_i \in \Sigma(y_i)$ for $i = 1, 2$, $M(a_1) = M(a_2)$ implies $\forall y_1' \in \beta(y_1, a_1)$, $\exists y_2' \in \beta(y_2, a_2)$ s.t. $(y_1', y_2') \in \Phi$,
   (If indistinguishable events a_i is defined at y_i $(i = 1, 2)$, then for each a_i-successor y_i' of y_i, exists a a_j-successor y_j' of y_j $(i, j = 1, 2)$, such that $(y_i', y_j') \in \Phi$.)
3) $y_1 \in Y_m$ implies $[y_2 \in Y_m] \vee [X_{syn}(y_2) \cap X_m = \emptyset]$.
   (The combined state should not change the marking status of the controlled system.)

Definition 5: Given G and S, S is *relative M-compatible* if exists a relative M-compatible bisimulation relation Φ over states of S such that $(y_0, y_0) \in \Phi$ for all $y_0 \in Y_0$.

Remark 1: Relative M-compatible bisimulation relation is closed under intersection. To see this, consider two relative M-compatible bisimulation relations Φ_1 and Φ_2 over states

of S. Pick any state-pair $(y_1, y_2) \in \Phi_1 \cap \Phi_2$. By Definition 4, condition 1, 2 and 3 hold for (y_1, y_2). Thus, $\Phi_3 = \Phi_1 \cap \Phi_2$ is also a relative M-compatible bisimulation relation. Further if both Φ_1 and Φ_2 contain (y_0, y_0) for each $y_0 \in Y_0$, then so does $\Phi_3 = \Phi_1 \cap \Phi_2$. Therefore, whenever S is relative M-compatible, there always exists a desired relative M-compatible bisimulation relation that is unique in the sense that it is the minimum one; which is what we work with by default.

The following lemma proves our earlier conjecture that M-compatibility of S and determinism of G implies relative M-compatibility of $G\|S$.

Lemma 1: Given deterministic G and M-compatible S, $G\|S$ is relative M-compatible.

The following example substantiates our earlier claim that relative M-compatibility is not preserved under bisimilarity, and as a result M-compatibility of S only guarantees relative M-compatibility of $G\|S$, and not of a specification R that is bisimilar to $G\|S$.

Example 2: Consider an example shown in Figure 2, where $M(a_1) = M(a_2)$ and identity mask for other events. It can be easily verified that $R_1 \simeq R_2$, R_1 is relative M-

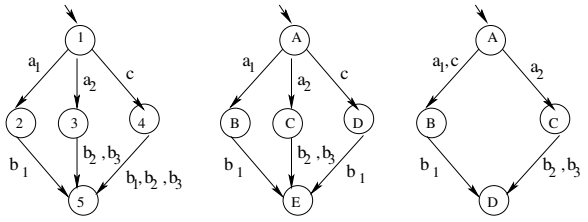

Fig. 2. G (left), R_1 (middle), and R_2 (right)

compatible, but R_2 is not. To see this, consider state A in R_2. Since $M(a_1) = M(a_2)$, for R_2 to be relative M-compatible, condition 1 in Definition 4 should hold for the corresponding successors B and C. Since $X_{syn}(B) = \{2, 4\}$ and $[\Sigma(4) \cap \Sigma(C) = \{b_2, b_3\}] \not\subseteq [\Sigma(4) \cap \Sigma(B) = \{b_1\}]$, condition 1 does not hold. Thus, R_2 is not relative M-compatible.

Loss of relative M-compatibility under bisimilarity transformation motivates the following weaker notion.

Definition 6: Given G, R is said to be *state-recognizable (SR)* with respect to G if exists relative M-compatible S such that $S \simeq R$.

It is evident from the above definition that SR is weaker than relative M-compatibility and also that it is preserved under bisimilarity.

IV. Bisimilarity Control under Partial Observation

In this section we establish that the property of SR is an additional property required of a specification R (besides being G-simulated and SC) for the existence of a bisimilarity enforcing supervisor when plant is deterministic.

We need to first establish the main property of relative M-compatibility: That given a relative M-compatible S possessing an underlying relative M-compatible bisimulation relation Φ, it is possible to compute S^Φ such that S^Φ is M-compatible and $G\|S \simeq G\|S^\Phi$. Note that this result does not require that the plant be deterministic. The following algorithm computes S^Φ by combining relative M-compatible bisimilar states.

Definition 7: Suppose S is relative M-compatible, possessing an underlying relative M-compatible bisimulation relation Φ. We say that $\hat{Y} \subseteq Y$ is a Φ-*compatible set* if $y_1, y_2 \in \hat{Y}$ implies $(y_1, y_2) \in \Phi$.

Note that by definition Φ is symmetric, and also contains (y_0, y_0) for all $y_0 \in Y_0$. Using the definition of relative M-compatibility it can then be concluded that $(y, y) \in \Phi$ for each $y \in Y$, i.e., Φ is also reflexive. However Φ need not be transitive, and as a result, the Φ-compatible sets do not form a partition of Y, rather only a cover. The states of S^Φ are then chosen to be the maximal Φ-compatible sets.

Algorithm 1: Given a relative M-compatible bisimulation relation Φ under which S is relative M-compatible, the algorithm for computation of a M-compatible S^Φ is given as below.

$$S^\Phi := (\mathcal{Y}, \Sigma, \beta^\Phi, \mathcal{Y}_0, \mathcal{Y}_m),$$

where

- $\mathcal{Y} \subseteq 2^Y$ is its set of states of S^Φ, and
$$\mathcal{Y} = \{\hat{Y} \subseteq Y \mid \hat{Y} \text{ is a maximal } \Phi\text{-compatible set}\}.$$
- β^Φ is its transition function, and for $\hat{Y}, \overline{Y} \in \mathcal{Y}$ and
$$\sigma \in \Sigma(\hat{Y}, \overline{Y}) = \{\sigma \in \Sigma(\hat{Y}) \mid \beta(\hat{Y}, \sigma) \cap \overline{Y} \neq \emptyset\},$$
where $\Sigma(\hat{Y}) = \cup_{y \in \hat{Y}} \Sigma(y)$,
$$\overline{Y} \in \beta^\Phi(\hat{Y}, \sigma) \Leftrightarrow M^{-1}M(\sigma) \cap \Sigma(\hat{Y}) \subseteq \Sigma(\hat{Y}, \overline{Y}).$$
- $\mathcal{Y}_0$ is its set of initial states, and $\mathcal{Y}_0 = \{\hat{Y} \in \mathcal{Y} \mid \hat{Y} \cap Y_0 \neq \emptyset\}$.
- $\mathcal{Y}_m$ is its set of marked states, and $\mathcal{Y}_m = \{\hat{Y} \in \mathcal{Y} \mid \hat{Y} \cap Y_m \neq \emptyset\}$.

The following theorem proves the correctness of Algorithm 1.

Theorem 1: Algorithm 1 is correct. I.e., consider G, S and an observation mask M. Suppose S is relative M-compatible so that exists a relative M-compatible bisimulation relation Φ such that $(y_0, y_0) \in \Phi$ for all $y_0 \in Y_0$. Then S^Φ is M-compatible, where S^Φ is computed by Algorithm 1.

Remark 2: Note that the state set of S^Φ is a subset of the power set 2^Y. In other words, the complexity of Algorithm 1 is in general exponential in the number of states of S, i.e., $O(2^{|S|})$.

Having showed that S^Φ is M-compatible whenever S is relative M-compatible, we next show that if in addition S is bisimilarity enforcing, then so is S^Φ.

Theorem 2: Given G and S, if S is relative M-compatible with respect to G, then $G\|S \simeq G\|S^\Phi$, where Φ is the relative M-compatible bisimulation relation under which S is relative M-compatible, and S^Φ is computed by Algorithm 1.

So far we have shown that a relative M-compatible state machine may be converted into a M-compatible state machine while preserving the control action. We next show

that if the state machine is also state-controllable, then it may be converted to a (Σ_u, M)-compatible state machine, also preserving the control action.

We know that if S is relative M-compatible then S^{Φ} is M-compatible. We show that if S is also SC, then so is S^{Φ}.

Lemma 2: Consider G and S such that S is relative M-compatible and SC. Then S^{Φ} is SC, where Φ is the relative M-compatible bisimulation relation under which S is relative M-compatible, and S^{Φ} is computed by Algorithm 1.

Finally, we show that S^{Φ} can further be made Σ_u-compatible by augmenting each state of S^{Φ} by undefined uncontrollable events at that state such that the control exercised remains unaffected. (Note that since S^{Φ} is SC, adding transitions at each state on undefined uncontrollable events in S^{Φ} does not change the result of synchronous composition with G.)

Theorem 3: Given a SC and relative M-compatible state machine S, consider S^{Φ} where Φ is the relative M-compatible bisimulation relation under which S is relative M-compatible, and S^{Φ} is as computed by Algorithm 1. For each state $\hat{Y}$ of S^{Φ} and $\sigma \in \Sigma_u - \Sigma(\hat{Y})$, add the following transition(s) on σ:

- If $\nexists a \in \Sigma(\hat{Y})$ such that $M(a) = M(\sigma)$, then add σ as self-loop at $\hat{Y}$;
- Else, $\forall a \in \Sigma(\hat{Y})$ such that $M(a) = M(\sigma)$, add transition on σ from $\hat{Y}$ to $\overline{Y} \in \beta^{\Phi}(\hat{Y}, a)$.

Let S^{Φ,Σ_u} be the resulting state machine. Then S^{Φ,Σ_u} is (Σ_u, M)-compatible, and $G \| S \simeq G \| S^{\Phi,\Sigma_u}$.

Now we are ready to establish a necessary and sufficient condition for the existence of a bisimilarity enforcing supervisor when plant is deterministic.

Theorem 4: Consider a deterministic G, a possibly nondeterministic R, and an observation mask M. Then there exists a (Σ_u, M)-compatible supervisor S such that $G \| S \simeq R$ if and only if $R \sqsubseteq G$, R is SC and SR.

V. On Verification of State Recognizability

In order to verify the existence of a bisimilarity enforcing supervisor for a deterministic plant using Theorem 4, we need a method to verify the SR property of R. (Methods to verify $R \sqsubseteq G$ are well-known, and a method to verify SC property of R was reported in [9].)

The following theorem establishes a way to verify state-recognizability of a specification R that is G-simulated.

Theorem 5: Given a deterministic G and a nondeterministic R such that $R \sqsubseteq G$, R is SR if and only if $G \| R$ is relative M-compatible.

Remark 3: According to Theorem 4, S^{Φ,Σ_u} can be used as a supervisor, where S is any state machine such that $S \simeq R$ and S is relative M-compatible. According to Theorem 5, S can be chosen to be $G \| R$. Thus a possible (Σ_u, M)-compatible bisimilarity enforcing supervisor for a deterministic plant is given by $(G \| R)^{\Phi,\Sigma_u}$. The complexity of its computation is same as that of $(G \| R)^{\Phi}$ since the complexity of computing $(\cdot)^{\Sigma_u}$ is linear in size of $(\cdot)$. Recall that a state in $(G \| R)^{\Phi}$ is a subset of $X \times Q$ implying that the complexity of synthesizing a bisimilarity enforcing supervisor for a deterministic plant is of order $O(2^{|G| \times |R|})$.

Next we illustrate Theorem 4 by revisiting Example 1.

Example 3: Applying Theorem 4, we first check whether R is G-simulated, which can be easily verified. Also, since $\Sigma_u = \emptyset$, R is trivially SC.

Next, we verify whether $G \| R$ is relative M-compatible. We find the following relative M-compatible bisimulation relation $\Phi \subseteq (X \times Q)^2$ exists (the details are omitted):

$$\begin{aligned}\Phi \;=\; & \{((A,1),(A,1)),((B,2),(C,10)),\\ & ((B,3),(C,9)),((D,4),(E,12)),\\ & ((D,5),(E,11)),((F,6),(G,14)),\\ & ((F,7),(G,13)),((H,8),(K,15)),\\ & ((C,10),(B,2)),((C,9),(B,3)),\\ & ((E,12),(D,4)),((E,11),(D,5)),\\ & ((G,14),(F,6)),((G,13),(F,7)),\\ & ((K,15),(H,8))\}.\end{aligned}$$

Thus, R is SR. Therefore, there exists a (Σ_u, M)-compatible supervisor S such that $G \| S \simeq R$.

We compute $(G \| R)^{\Phi}$ by Algorithm 1. We have

$$\begin{aligned}\mathcal{Y} \;=\; & \{\{(A,1),(A,1)\},\{(B,2),(C,10)\},\\ & \{(B,3),(C,9)\},\{(D,4),(E,12)\},\\ & \{(D,5),(E,11)\},\{(F,6),(G,14)\},\\ & \{(F,7),(G,13)\},\{(H,8),(K,15)\}\},\end{aligned}$$

and $\mathcal{Y}_0 = \{\{(A,1),(A,1)\}\}$.

Next we compute the set of transitions. Let

$$\begin{aligned}&\hat{Y}_0 = \{(A,1),(A,1)\}, \hat{Y}_1 = \{(B,2),(C,10)\},\\ &\hat{Y}_2 = \{(B,3),(C,9)\}, \hat{Y}_3 = \{(D,4),(E,12)\},\\ &\hat{Y}_4 = \{(D,5),(E,11)\}, \hat{Y}_5 = \{(F,6),(G,14)\},\\ &\hat{Y}_6 = \{(F,7),(G,13)\}, \hat{Y}_7 = \{(H,8),(K,15)\}.\end{aligned}$$

Note that $\Sigma(\hat{Y}_0) = \{a_1, a_2\}$ and $\Sigma(\hat{Y}_1) = \{b_1, b_3\}$.

- Since $\Sigma(\hat{Y}_0, \hat{Y}_1) = \{a_1, a_2\}$ and $M^{-1}M(a_1) \cap \Sigma(\hat{Y}_0) = \{a_1, a_2\} \subseteq \Sigma(\hat{Y}_0, \hat{Y}_1)$, transitions $(\hat{Y}_0, a_1, \hat{Y}_1)$ and $(\hat{Y}_0, a_2, \hat{Y}_1)$ are in $(G \| R)^{\Phi}$.
- Since $\Sigma(\hat{Y}_1, \hat{Y}_3) = \{b_1, b_3\}$ and $M^{-1}M(b_1) \cap \Sigma(\hat{Y}_1) = \{b_1, b_3\} \subseteq \Sigma(\hat{Y}_1, \hat{Y}_3)$, transitions $(\hat{Y}_1, b_1, \hat{Y}_3)$ and $(\hat{Y}_1, b_3, \hat{Y}_3)$ are in $(G \| R)^{\Phi}$.

Similarly one can compute the other transitions; the details are omitted here. $(G \| R)^{\Phi}$ is drawn in Figure 3.

Since $\Sigma_u = \emptyset$, S is same as $(G \| R)^{\Phi}$.

VI. Test for Relative M-compatibility

Since state-recognizability property of R is reduced to a relative M-compatibility property (under determinism of G and the fact that R is G-simulated), we next develop an algorithm that polynomially verifies relative M-compatibility.

Algorithm 2: The algorithm for verifying relative M-compatibility of S is given as below:

1) Consider two copies of S and perform their masked-composition, denoted $MC(S, S)$. (In MC, indistinguishable pair of events are synchronized)).

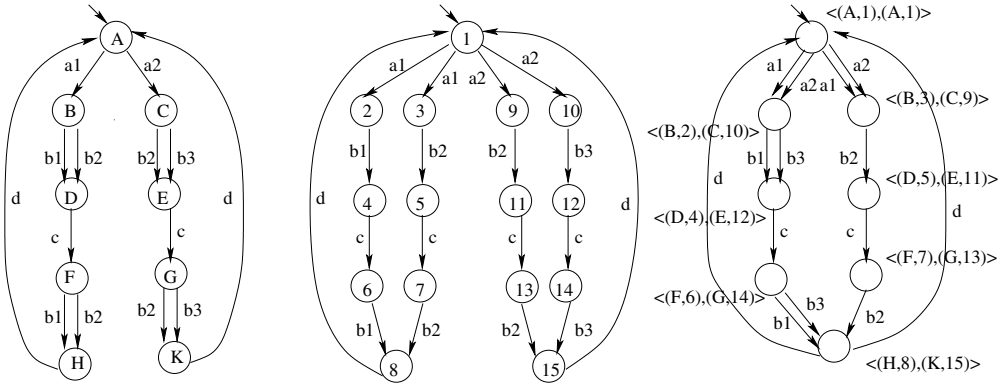

Fig. 3. G (left), R (middle), and S (right)

2) Mark a state (y_1, y_2) of $MC(S,S)$ "good" if and only if $\forall x_i \in X_{syn}(y_i), \Sigma(x_i) \cap \Sigma(y_j) \subseteq \Sigma(x_i) \cap \Sigma(y_i), i,j = 1,2$, and $y_i \in Y_m, y_j \notin Y_m \Rightarrow X_{syn}(y_j) \cap X_m = \emptyset, i,j = 1,2$.
3) a) $n := 0$,
$\Phi_n := \{(y_1,y_2) \mid (y_1,y_2)$
is a "good" state in $MC(S,S)\}$.
b) For $(y_1,y_2) \in \Phi_n$, if $a_i \in \Sigma(y_i)$ for $i = 1,2$, and $M(a_1) = M(a_2)$, then for each $y_i' \in \beta(y_i, a_i)$, check whether exists $y_j' \in \beta(y_j, a_j)$, where $i,j = 1,2$, such that $(y_1', y_2') \in \Phi_n$. If not, $\Phi_{n+1} := \Phi_n - \{(y_1,y_2)\}$.
c) If $\Phi_{n+1} = \Phi_n$ or $\Phi_{n+1} = \emptyset$, then $\Phi := \Phi_{n+1}$ and stop; Otherwise, $n := n+1$, go to step (b).
4) S is relative M-compatible if and only if $(y_0, y_0) \in \Phi$ for all $y_0 \in Y_0$.

The following theorem proves the correctness of Algorithm 2.

Theorem 6: Algorithm 2 is correct.

Remark 4: The complexity of masked-composition of S and S is $O(|S|^2)$. The complexity of marking "good" states is linear in the size of $G\|S$ and quadratic in the size of S, i.e., $O(|G| \times |S|) + O(|S|^2)$. The complexity of step 3 is linear in the size of $MC(S,S)$, i.e., $O(|S|^2)$. Thus, the complexity of Algorithm 2 is linear in the size of $G\|S$ and quadratic in the size of S, i.e., $O(|G| \times |S|) + O(|S|^2)$.

It follows that the complexity of checking relative M-compatibility of $G\|R$ is $O(|G| \times (|G| \times |R|)) + O((|G| \times |R|)^2) = O(|G|^2 \times |R|^2)$. Complexity of checking R is SC is $O(|G| \times |R|)$. Complexity for checking whether R is G-simulated is $O(|G| \times |R|)$. Thus, the complexity of checking the existence of a (Σ_u, M)-compatible bisimilarity enforcing supervisor for deterministic plants is $O(|G|^2 \times |R|^2)$.

VII. Conclusion

This paper studied the supervisory control of deterministic systems subject to nondeterministic specifications under partial observation, with the objective that the controlled system be bisimulation equivalent to the specification system. A main motivation for investigating the special case of deterministic plants is to determine whether in this case the problem has a more manageable complexity. The answer turns out to be positive. The existence condition we find is polynomially verifiable, whereas the complexity of synthesizing a supervisor (when one exists) is singly exponential. These complexity classes are similar to ones for control under partial observation in a deterministic setting [5]. We obtained a necessary and sufficient condition for the existence of a supervisor in terms the notion of state-recognizability introduced in this paper, and presented an algorithm of polynomial complexity for verifying it. The presence of partial observation poses a new challenge in the setting of nondeterministic specifications. It turns out certain properties that are relevant in the setting of partial observation such as observation-compatibility are not preserved under bisimilarity. An elaborate computation is required to show that the composition of the plant and the specification when suitably transformed through certain state-mergers can be used as a supervisor.

References

[1] R. Kumar, S. Jiang, C. Zhou, and W. Qiu. Polynomial synthesis of supervisor for partially observed discrete-event systems by allowing nondeterminism in control. *IEEE Transactions on Automatic Control*, 50(4):463–475, 2005.
[2] O. Kupferman, P. Madhusudan, P.S. Thiagarajan, and M.Y. Vardi. Open systems in reactive environments: Control and synthesis. In *Proc. 11th Int. Conf. on Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 92–107. Springer-Verlag, 2000.
[3] P. Madhusudan and P.S. Thiagarajan. Branching time controllers for discrete event systems. *Theoretical Computer Science*, 274:117–149, 2002.
[4] P. Tabuada. Open maps, alternating simulations and control synthesis. In *International Conference on Concurrency Theory*, pages 466–480, 2004.
[5] J. N. Tsitsiklis. On the control of discrete event dynamical systems. *Mathematics of Control Signals and Systems*, 2(2):95–107, 1989.
[6] C. Zhou and R. Kumar. Control of nondeterministic discrete event systems for simulation equivalence. In *Proceedings of IEEE Conference on Decision and Control*, Nassau, Bahama, 2004.
[7] C. Zhou and R. Kumar. Small model theorem for bisimilarity control under partial observation. In *Proceedings of American Control Conference*, 2005. Accepted.
[8] C. Zhou, R. Kumar, and S. Jiang. Control of nondeterministic discrete event systems for bisimulation equivalence. In *Proceedings of 2004 American Control Conference*, pages 4488–4492, Boston, MA, June 2004.
[9] C. Zhou, R. Kumar, and S. Jiang. Control of nondeterministic discrete event systems for bisimulation equivalence. *IEEE Transactions on Automatic Control*, 2004. Accepted.