A Small Model Theorem for Bisimilarity
Control under Partial Observation

Changyan Zhou, Member, IEEE, and Ratnesh Kumar, Senior Member, IEEE

Abstract— The paper extends our prior result on decidability of bisimulation equivalence control reported in [7], [8] from the setting of complete observations to that of partial observations. Besides being control-compatible the supervisor must now also be observation-compatible. We show that the “small model theorem” remains valid by showing that a control and observation compatible supervisor exists if and only if it exists over a certain finite state space, namely the power set of the Cartesian product of the system and the specification state spaces.

Note to Practitioners: Nondeterminism in discrete-event systems arises due to abstraction and/or unmodeled dynamics. The paper addresses the issue of control of nondeterministic systems subject to nondeterministic specifications, under a partial observation of events. Nondeterministic plant and specification are useful when designing a system at a higher level of abstraction so that lower level details of system and its specification are omitted to obtain higher level models that are nondeterministic. The control goal is to ensure that the controlled system has an equivalent behavior as the specification system, where the notion of equivalence used is that of bisimilarity. Bisimilarity requires the existence of an equivalence relation between the states of the two systems so that transitions on common events starting from a pair of equivalent states end up in a pair of equivalent successor states. Supervisors are also allowed to be nondeterministic, where the nondeterminism in control is implemented by selecting control actions nondeterministically from a set of pre-computed choices (for more detail see [2].) The main contribution of the paper is to show that a supervisor exists under partial observation for achieving bisimilarity for nondeterministic systems. Section 4 gives an illustrative example. The paper concludes with Section 5.

II. NOTATION AND PRELIMINARIES

We model DESs as nondeterministic state machines (NSMs). A NSM $G$ is a five tuple: $G := (X, \Sigma, \alpha, X_0, X_m)$, where $X$ is its set of states, $\Sigma$ is its set of events, $\alpha : X \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^X$ 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. For an event set $\Sigma$, we use $\Sigma$ to denote $\Sigma \cup \{\epsilon\}$. $\Sigma^*$ denotes the set of all finite-length sequences of events from $\Sigma$. An element (resp., a subset) of $\Sigma^*$ is called a trace (resp., a language). The $\epsilon$-closure of $x \in X$, denoted as $\epsilon^*(x)$, is the set of states reached by the execution of zero or more $\epsilon$-transitions from state $x$. By using the $\epsilon$-closure map, we can extend the definition of transition function from events to traces, denoted $\alpha^* : X \times \Sigma^* \rightarrow 2^X$, which is defined inductively for each $x \in X$ as follows:

$$
\alpha^*(x, \epsilon) := \epsilon^*(x);
$$

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

where for $X \subseteq X$ and $\Sigma \subseteq \Sigma$, $\alpha^*(X, \Sigma) := \cup_{s \in X} \alpha^*(x, \sigma)$, and $\epsilon^*(X) := \cup_{x \in X} \epsilon^*(x)$. The language generated (resp., marked) by $G$, is denoted as $L(G)$ (resp., $L_m(G)$). $L(G)$ is the sequences of events executable starting from the initial state, i.e., $L(G) = \{s \in \Sigma^* | \alpha^*(X_0, s) \neq \emptyset\}$, and $L_m(G)$ is the set of generated sequences that end in a marked state, i.e., $L_m(G) = \{s \in L(G) | \alpha^*(X_0, s) \cap X_m \neq \emptyset\}$. For $x \in X$ we define $\Sigma(x) := \{\sigma \in \Sigma | \alpha^*(x, \sigma) \neq \emptyset\}$ 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 $G_i = (X_i, \Sigma, \alpha_i, X_{i0}, X_{im})$, is the automaton

$$
G_1 \parallel G_2 = (X_1 \times X_2, \Sigma, \alpha_{\parallel}, X_{10} \times X_{02}, X_{m1} \times X_{m2}),
$$

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

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

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

$$
G_1 \cup G_2 = (X_1 \cup X_2, \Sigma, \alpha_u, X_{10} \cup X_{02}, X_{m1} \cup X_{m2}),
$$

where for $x \in X_1 \cup X_2$, $\sigma \in \Sigma$

$$
\alpha_u(x, \sigma) := \begin{cases}
\alpha_1(x, \sigma) \cup \alpha_2(x, \sigma) & \text{if } x \in X_1 \cap X_2 \\
\alpha_1(x, \sigma) & \text{if } x \in X_1 - X_2 \\
\alpha_2(x, \sigma) & \text{if } x \in X_2 - X_1
\end{cases}
$$
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 \( M : \Sigma \rightarrow \overline{\Delta} \) (\( \Delta \) is the set of observed symbols), satisfying \( M(\varepsilon) = \varepsilon, \sigma \in \Sigma \) is said to be an unobservable event if \( M(\sigma) = \varepsilon, \) and otherwise it is said to be an observable event. Two events \( \sigma_1, \sigma_2 \in \Sigma \) are said to be indistinguishable if \( M(\sigma_1) = M(\sigma_2). \) The observation mask \( M \) is extended to be defined over traces in \( \Sigma^* \) as follows: \( M(\varepsilon) := \varepsilon; \ \forall \sigma \in \Sigma, \sigma \in \Sigma : M(\sigma) := M(\sigma) \cdot M(\varepsilon). \) Bisimulation equivalence is a type of behavioral equivalence that is used to describe equivalence between nondeterministic systems, and requires the existence of a bisimulation relation over the states of the two systems. A bisimulation relation is a symmetric simulation relation, which is defined as follows.

**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 all \( x_1, x_2 \in X_1 \cup X_2, \) \( (x_1, x_2) \in \Phi \) implies

1. \( \sigma \in \Sigma, \ \exists \bar{\sigma} \in \alpha_2(x_1, \sigma) \Rightarrow \exists x_2' \in \alpha_1(x_2', \sigma) \text{ such that } (x_2', x_2) \in \Phi. \)
2. \( x_1 \in X_{01} \cup X_{m1} \Rightarrow x_2 \in X_{m1} \cup X_{02}. \)

\( G_1 \) 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 \( x_{01} \in X_{01}, \) there is a \( x_{02} \in X_{02} \) 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 \( \Phi \) with \( (x_1, x_2) \in \Phi, \) read as \( x_1 \) is simulated by \( x_2. \) We sometimes omit the subscript \( \Phi \) when it is clear from the context. A simulation relation is called a bisimulation relation if it is symmetric. For a bisimulation relation \( \Phi \) if \( (x_1, x_2) \in \Phi, \) then \( x_1 \) and \( x_2 \) are called bisimilar, written as \( x_1 \approx_{\Phi} \) \( x_2 \) (or simply \( x_1 \approx \) \( x_2 \) when \( \Phi \) is clear from context). Two automata \( G_1 \) and \( G_2 \) are said to be bisimilar, denoted as \( G_1 \approx_{\Phi} G_2, \) if there exists a bisimulation relation \( \Phi \) such that for all \( x_{01} \in X_{01}, \) there is a \( x_{02} \in X_{02} \) such that \( x_{01} \approx_{\Phi} x_{02}, \) denoted for short as \( X_{01} \approx_{\Phi} X_{02}. \)

**III. Bisimilarity Control under Partial Observation**

A supervisor observes events through an observation mask \( M. \) If two events are observationally indistinguishable, and are enabled at a state, then a supervisor must perform identical state update on them, i.e., the corresponding successor states be the same. Also a state remains unaltered when an unobservable event is executed. This requirement is referred to as \( M.\)-compatibility. Events in a set \( \Sigma_u \subseteq \Sigma \) are uncontrollable and must never be disabled by a supervisor. So besides being \( M.\)-compatible (to accommodate limitations of partial observability) a supervisor must also be \( \Sigma_u.\)-compatible, namely, all uncontrollable events be always enabled. We will 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.

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

- \( S \) is called \( \Sigma_u.\)-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 \( \forall a \in \Sigma(y), \) if \( M(a) = M(b), \) then \( \beta(y, a) = \beta(y, b); \) and \( \forall c \in \Sigma(y), \) if \( M(c) = c, \) then \( \beta(y, c) = \{ y \}. \)
- \( S \) is called \( (\Sigma_u, M)\)-compatible if \( S \) is \( \Sigma_u\)-compatible and \( M\)-compatible.

The small model theorem in [8] states that a bisimilarity enforcing \( \Sigma_u\)-compatible supervisor exists if and only if it exists over the state space \( 2^X \times Q, \) where \( X \) is the state space of the plant \( G \) and \( Q \) is the state space of the specification \( R. \) The sufficiency is clearly obvious, while the key idea behind the necessity is that given a \( \Sigma_u\)-compatible bisimilarity enforcing supervisor \( S \) (i.e., \( G\| S \cong R \)), each state \( y \) in \( Y \) of \( S \) can be labeled by \( lbl(y) \subseteq X \times \Sigma \), and then the states carrying the identical labels can be merged to obtain a state machine \( T \) with state space \( 2^X \times \Sigma \) such that \( T \) is \( \Sigma_u\)-compatible and \( G\| T \cong R \). We recall from [8] that \( (x, q) \in X \times Q \) belongs to \( lbl(y) \) if and only if \( (x, y) \) is a state in \( G\| S \) that is bisimilar to the state \( q \) of \( R. \) In other words, \( lbl(y) = \{ (x, q) \in X \times Q | (x, y) \cong q \}. \) We use the same labeling function for extending the small model theorem to the setting of partial observation.

The following example illustrates how the labels are computed.

**Example 1:** Consider \( G, S \) and \( R \) shown in Figure 1. It can be verified that \( G\| S \cong R \) since the following bisimulation relation exists between \( G\| S \) and \( R \) (for the sake of simplicity we write a state \( (x, y) \) as \( x y)\))

\[
\Phi := \{ (x_0 y_0, y_0), (x_1 y_1, x_1), (x_1 y_2, q_2), (x_2 y_3, q_3), (x_3 y_4, q_4), (x_4 y_5, q_5), (x_5 y_0, q_0), (x_1 y_1, x_1), (q_2, x_1 y_2), (q_3, x_2 y_3), (q_4, x_3 y_4), (q_4, x_4 y_5), (q_5, x_4 y_8), (q_1, x_1 y_6), (q_2, x_1 y_7), (q_3, x_2 y_8), (q_4, x_3 y_9), (q_4, x_4 y_8) \}.
\]

Using this bisimulation relation, we construct Table I in order to compute the labeling function of each state in \( S. \) Entries in the topmost row (resp., leftmost column) represent a state \( x \) in \( G \) (resp., \( y \) in \( S \)), and the corresponding cell entry is a state \( q \) in \( R \) such that \( q \cong_{\Phi} (x, y). \) \( lbl(y) \subseteq X \times Q \) contains all \( (x, q) \) pairs such that \( q \cong_{\Phi} (x, y). \) Such a labeling is depicted in Figure 2.

Since \( lbl(y_1) = lbl(y_6) \) and \( lbl(y_2) = lbl(y_7) \), we merge \( y_1 \) and \( y_6, \) and \( y_2 \) and \( y_7, \) respectively. The state machine \( T \) obtained by merging states in \( S \) carrying the merged state label is also drawn in Figure 2. It can be seen that \( G\| T \cong R. \)
The following theorem establishes the small model theorem for bisimilarity enforcing control under partial observability of events.

**Theorem 1:** Given G and R, and a mask M, there exists a\((Σ_u, M)\)-compatible supervisor S such that \(G|S \simeq R\) if and only if there exists a \((Σ_u, M)\)-compatible state machine T with state space \(2^{X \times Q}\) such that \(G|T \simeq R\).

**Proof:** *If* For necessity, suppose exists \((Σ_u, M)\)-compatible S such that \(G|S \simeq R\). Without loss of generality, each non-epsilon transition of S synchronizes with some transition of G when the two systems are composed (otherwise we can simply omit such transitions from S). Label each \(y \in Y\) of S by \(lb(l) \subseteq X \times Q\) where \((x, q) \in lb(l)\) if and only if \((x, y)\) reachable in \(G|S\) and \(q \in Q\) is such that \((x, y) \simeq q\). Merge all states carrying the same label, and call the resulting state machine T. Then from proof of [8, Theorem 2], \(G|T \simeq R\), T is \(Σ_u\)-compatible, and state space of T is \(2^{X \times Q}\). We claim that T is also \(M\)-compatible, which will prove the necessity. Suppose \(y_1, y_2 \in Y\) are such that \(lb(l) = lb(y_2)\). Then \(y_1\) and \(y_2\) are merged to obtain the state denoted \((y_1, y_2)\). Using the fact that each non-epsilon transition of S synchronizes with some transition of G, it was shown in proof of [8, Theorem 2] that,

\[
\Sigma(y_1) = \bigcup_{(x, q) \in lb(l)} \Sigma(q) = \bigcup_{(x, q) \in lb(y_2)} \Sigma(q) = \Sigma(y_2).
\]

So after merger, \(\Sigma((y_1, y_2)) = \Sigma(y_1) = \Sigma(y_2)\). Since S is \(M\)-compatible, for any pair of indistinguishable events \(a_1, a_2 \in \Sigma(y_1) = \Sigma(y_2), \beta(y_1, a_1) = \beta(y_1, a_2)\) and \(\beta(y_2, a_1) = \beta(y_2, a_2)\). So

\[
\beta((y_1, y_2), a_1) = \beta(y_1, a_1) \cup \beta(y_2, a_1) = \beta(y_1, a_2) \cup \beta(y_2, a_2) = \beta((y_1, y_2), a_2).
\]

Further, since S is \(M\)-compatible, if \(\sigma \in \Sigma(y_1) = \Sigma(y_2)\) is such that \(M(\sigma) = \epsilon\), then \(\beta(y_1, \sigma) = \{y_i\}\) for \(i = 1, 2\). It follows that \(\beta((y_1, y_2), \sigma) = \{y_i\}\). Thus T is also \(M\)-compatible.

Thus the merger of states carrying the same label preserves \(M\)-compatibility, and so T is \(M\)-compatible.

**(I)** Set \(S := T\), then S is \((Σ_u, M)\)-compatible and \(G|S \simeq G|T \simeq R\). This completes the proof.

**Remark 1:** From Theorem 1, an exhaustive search can be performed to determine the existence of a supervisor S over the state space \(2^{X \times Q}\) with the upper bound complexity of which is \(O(2^{2^{|X| \times |Q|}})\). From this, the upper bound complexity of checking the existence of a supervisor under partial observation is same as the one under full observation. Better upper bounds may exist, but are not known at this time.

**Remark 2:** A supervisor enforcing a bisimulation equivalence specification may be nondeterministic. Such a supervisor is implemented by selecting control actions nondeterministically from among a set of pre-computed choices (for more detail see [2]).

### IV. AN ILLUSTRATIVE EXAMPLE

To illustrate the bisimilarity control under partial observability, we present the following simple manufacturing example.

**Example 2:** Consider a manufacturing system (shown in Figure 3) consisting of two workstations, one robot and three storage-stations.
The robot moves among the workstations and storage-stations on guided rails. Initially, the robot departs from workstation 1 and nondeterministically travels on one of the rails (event $a$). On rail 1, the robot picks up a part from storage-station 1 (event $b_1$) and then delivers this part to workstation 2 for processing (event $c$). After the processing, robot returns the part to storage-station 1 (event $b_2$). On rail 2, the robot either picks up a part from storage-station 2 (event $b_3$) or from storage-station 3 (event $b_4$), and then delivers the part to workstation 2 for processing (event $c$). After the processing, the robot returns the part to either storage-station 2 or 3 (event $b_3$ or $b_4$). Not returning the part to its original storage-station is undesirable. After returning the part to the storage-station, the robot goes back to workstation 1 (event $a$) from where the entire process may be repeated. The state machine model $G$ of the system is drawn in Figure 4.

The specification $R$, also drawn in Figure 4, shows the acceptable behavior. According to the specification, the robot never picks a part from storage-station 2, and also returns any processed part to the same workstation from where it picked that part. A part once picked must be delivered to workstation 2 for processing, i.e., the event $c$ is uncontrollable. Only the events $a$ and $c$ are completely observable. Events $b_1, b_2$ and $b_3$ are observationally indistinguishable. Thus, we have $\Sigma = \{a, b_1, b_2, b_3, c\}$, $\Sigma_u = \{c\}$, and the observation mask $M$ is given by $M(a) = a$, $M(b_1) = M(b_2) = M(b_3) \neq c$ and $M(c) = c$. The control goal is to find a $(\Sigma_u, M)$-compatible supervisor $S$ such that the controlled system $G\|S$ is bisimilar to the specification $R$

Such a supervisor is drawn in Figure 5. Since $c$ is defined at each state of $S$, $S$ is $\Sigma_u$-compatible. Also state updates on indistinguishable pair of events $b_1$ and $b_2$ at states $y_1, y_3, y_5, y_6$, where they are both defined, are identical, implying that $S$ is also $M$-compatible.

The controlled system $G\|S$ is also drawn in Figure 5. The following bisimulation relation $\Phi$ exists between $G\|S$ and $R$:

$\Phi = \{(x_0 y_0, q_0), (x_1 y_1, q_1), (x_2 y_2, q_2), (x_3 y_3, q_3), (x_4 y_4, q_4), (x_5 y_5, q_5), (x_6 y_6, q_6), (x_7 y_7, q_7), (q_0, x_0 y_0), (q_1, x_1 y_1), (q_2, x_2 y_2), (q_3, x_3 y_3), (q_4, x_4 y_4), (q_5, x_5 y_5), (q_6, x_6 y_6), (q_7, x_7 y_7)\}$

From Theorem 1, exists a $(\Sigma_u, M)$-compatible $T$ with state space $2^{\times Q}$ such that $G\|T \simeq R$. To obtain such a $T$, the labeling of each state in $S$ is shown in Figure 6, and is computed using Table II. State machine $T$ is obtained by merging the states in $S$ carrying the same label. Since $lbb(y_1) = lbb(y_5)$, we merge $y_1$ and $y_5$. The resulting state machine $T$ is drawn in Figure 6. $T$ is $(\Sigma_u, M)$-compatible as expected. Moreover, $G\|T \simeq G\|S \simeq R$, where $G\|T$ is drawn in Figure 6.

**V. Conclusion**

In this note we studied the supervisory control of nondeterministic systems subject to nondeterministic specifications under partial observation, with the objective that the controlled system be bisimulation equivalent to the specification. We obtained a small model theorem showing that a control and observation compatible bisimilarity enforcing 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 state spaces, thereby proving the decidability of bisimilarity enforcing control under partial observation of general nondeterministic systems and nondeterministic specifications. It was shown in [7], [8] that in the special case when the plant is deterministic, the bisimilarity enforcing control under complete observation is polynomially solvable. The existence can be verified linearly in the size of the plant and the specification, and synthesis can be performed linearly in the size of the specification. It will be interesting to see if similar polynomial complexity results exist even for a partially observed plant when it is deterministic.

**References**

Fig. 6. The labeling of states in $S$ (left), $T$ (middle), and $G\|T$ (right)


