Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Control using Nondeterministic Supervisors for Partially Observed Discrete Event Systems

Ratnesh Kumar[a], Shengbing Jiang[b], Changyan Zhou[a], and Wenbin Qiu[a]
[a]Department of ECE, Iowa State University, Ames, IA 50011
[b]GM R&D and Planning, MC: 480-106-200, Warren, MI 48090-9055

***Abstract*— We study the supervisory control of discrete event systems under partial observation using *nondeterministic* supervisors. We formally de£ne a nondeterministic control policy and also a control & observation compatible nondeterministic state machine and prove their equivalence. We show that when control is exercised using a nondeterministic supervisor, the speci£cation language is required to satisfy a weaker notion of observability, which we de£ne in terms recognizability and achievability. Achievability serves as necessary and suf£cient condition for the existence of a nondeterministic supervisor, and it is weaker than controllability and observability combined. When all events are controllable, achievability reduces to recognizability. We show that both existence, and synthesis of nondeterministic supervisors can be determined polynomially. (For deterministic supervisors, only existence can be determined polynomially.) Both achievability and recognizability are preserved under union, also and under intersection (when restricted over pre£x-closed languages). Using the intersection closure property we derive a necessary and suf£cient condition for the range control problem for the pre£x-closed case. Unlike the deterministic supervisory setting where the complexity of existence is exponential, our existence condition is polynomially veri£able, and also a supervisor can be polynomially synthesized.**

Keywords: Discrete event system, Nondeterministic supervisory control, Partial observation, Controllability, Recognizability, Achievability

I. INTRODUCTION

Discrete event systems (DESs) are event-driven systems possessing discrete states that change when events occur. Supervisory control theory for DESs was proposed by Ramadge-Wonham [10]. Under a complete observation of events, the controllability of the desired behavior serves as a key property for the existence of a supervisor that can enforce the desired speci£cation.

The extension of supervisory control theory to deal with partial observability of events was presented in [9], [2]. It was shown that the additional property of observability plays an equal role in the existence of a supervisor enforcing the speci£cation. The property of observability can be tested polynomially in the size of the states of the plant and the speci£cation [13]. However, even when the observability holds, an off-line computation of the supervisor for control under partial observation has an exponential complexity [13]. For pre£x-closed speci£cations, a procedure for the off-line computation of a maximally permissive supervisor is reported in [1], and an algorithm for the on-line computation of a maximally permissive supervisor, possessing a polynomial step-wise complexity, is reported in [3]. One issue with the computation of a maximally permissive supervisor for control under partial observation is that the property of observability is preserved only over an increasing chain, but not in general [9]. So while a unique supremal observable sublanguage does not exist, maximal observable sublanguages exist, which is what maximally permissive supervisors attain.

The reason for (i) the exponential complexity of the off-line computation of the supervisor for control under partial observation when observability holds, and (ii) the non-existence of a unique maximally permissive supervisor is the underlying requirement that the supervisor be *deterministic*, where the next control action is uniquely determined as a function of the history of the observations. There is no reason for having this underlying restriction of determinism, and in fact in control of stochastic systems both deterministic and randomized control policies are used [6].

In this paper, we formally de£ne a nondeterministic control policy and also a control & observation compatible nondeterministic state machine and prove their equivalence. We show that a nondeterministic control policy can be concisely represented as a control & observation compatible nondeterministic state machine. When control is exercised using a nondeterministic supervisor, the speci£cation language is required to satisfy a weaker notion of observability, which we de£ne in terms recognizability and achievability. Achievability servers as a necessary and suf£cient condition for the existence of a nondeterministic supervisor, and it is weaker than controllability and observability combined. When all events are controllable, achievability reduces to recognizability. I.e., the recognizability captures the restriction arising from observational limitations, whereas the achievability captures the restriction caused by the combined control and observation limitations. We further show that both existence, and synthesis of nondeterministic supervisors can be done polynomially. (As mentioned above, for deterministic supervisors, only existence can be determined polynomially.) Furthermore, both achievability and recognizability are preserved under union, which are

This work was supported in part by the National Science Foundation under the grants NSF-ECS-0218207, NSF-ECS-0244732 and NSF-EPNES-0323379, and a DoD-EPSCoR grant from the Of£ce of Naval Research under the grant N000140110621.

their another advantage over observability. Consequently, a unique maximally permissive supervisor for control under partial observation exists. Although the computation of the maximally permissive supervisor is beyond the scope of this paper, we have shown its existence by showing the closure of achievability under set union.

Inan [4], [5] £rst advocated the use of nondeterministic supervisors for control under partial observation. In his study, nondeterministic supervisors were restricted to have no silent or ϵ-transitions, and partial observation was restricted to be a projection type observation mask. Inan introduced the notion of weakly controllable and observable languages for the characterization of languages achievable under nondeterministic supervisors. The class of such languages was shown to be closed under union, and an algorithm of exponential complexity for computing the supremal element was provided. The notion of achievability de£ned in this paper can be viewed as a generalization of the weak controllability and observability to allow for the silent transitions in the supervisors and also non-projection type observation masks. Inclusion of silent transitions makes the supervisors more general. Achievability is a weaker notion than the weak controllability and observability. Moreover, our characterization of the properties of recognizability and achievability has quite a different form that lets us separate the observation issues from the control issues. Finally, the form in which we de£ne the properties of recognizability and achievability, it lends to an easy polynomial complexity veri£cation algorithm. In contrast, the weak controllability and observability veri£cation issue was not explicitly addressed in [5]. One could compute the supremal sublanguage using the algorithm given in [5] and check it's equality with the language itself, but that would have a double exponential complexity.

In [12], it was studied how two "process-objects" interacting via "masked-composition" can achieve a certain closed-loop language. The logic of process-objects was represented as nondeterministic state machines, but the work did not explicitly explore any issue pertaining to nondeterministic control.

While observability is not preserved under union, it is preserved under intersection when restricted over pre£x-closed languages [9]. As a result a unique in£mal pre£x-closed and observable superlanguage of the speci£cation language exists. Its computation is useful in the solution of pre£x-closed "range" control problem where the controlled behavior is required to lie in a speci£ed range given as upper and lower bound pre£x-closed languages. The computation of in£mal pre£x-closed and observable superlanguage is reported in [11] and that of the in£mal pre£x-closed controllable and observable superlanguage in [7], [8]. In this paper we show that alike observability and controllability, recognizability and achievability are preserved under intersection (when restricted over pre£x-closed languages). Using the intersection closure property we derive a necessary and suf£cient condition for the range control problem for the pre£x-closed case. Unlike the deterministic supervisory setting where the complexity of existence is exponential [13], our existence condition is polynomially veri£able, and also a supervisor can be polynomially synthesized.

II. NONDETERMINISTIC CONTROL

In this section we formally de£ne a nondeterministic control policy, and a control & observation compatible nondeterministic supervisor state machine. We prove that given a nondeterministic control policy it can be represented as a control & observation compatible supervisor state machine, and vice-versa, in the sense that the two achieve the same class of languages as the closed-loop behavior.

A. Nondeterministic Control Policy

We use $\Sigma_u \subseteq \Sigma$ to denote the set of uncontrollable events, and $\Gamma = \{\hat{\Sigma} \subseteq \Sigma \mid \hat{\Sigma} \supseteq \Sigma_u\}$ to denote the set of control actions. Letting $M : \overline{\Sigma} \to \overline{\Delta}$ with $M(\epsilon) = \epsilon$ denote the observation mask (we use $\overline{\Sigma}$ to denote $\Sigma \cup \{\epsilon\}$), we know that a deterministic control policy is a map $f : \Delta^* \to \Gamma$ that maps each history (a sequence of observation symbols) into a unique control action.

A nondeterministic control policy generalizes such a deterministic one in several ways:

- Instead of specifying a unique control action for each history, it speci£es a set of control actions, one of which is to be nondeterministically chosen at the run-time (at the time when control is being exercised). This is captured by having the range of f as 2^Γ (instead of Γ).
- Off-line a nondeterministic control policy speci£es a set of control action choices, one of which is selected nondeterministically on-line. So a history now consists of an observation-trace together with a trace of selected control actions. In other words, a history of length k may be of the form
$$d_k = (\hat{\Sigma}_1, \tau_1), \ldots, (\hat{\Sigma}_k, \tau_k) \in (\Gamma \times \Delta)^*,$$
where for $i \leq k$, $\hat{\Sigma}_i \in \Gamma$ is the ith control action selected, and $\tau_i \in \Delta$ is the ith observed symbol.
- A nondeterministic control policy can change the control action prior to the arrival of a new observation. This is captured by allowing τ_i to be ϵ in above, i.e., $\tau_i \in \Delta \cup \{\epsilon\} = \overline{\Delta}$. Thus the set of all possible histories is given by $(\Gamma \times \overline{\Delta})^*$.

De£nition 1: A non-deterministic control policy f over (Σ_u, M) is a partial function
$$f : (\Gamma \times \overline{\Delta})^* \to 2^\Gamma$$
such that $\forall d_k = (\hat{\Sigma}_1, \tau_1) \cdots (\hat{\Sigma}_k, \tau_k) \in (\Gamma \times \overline{\Delta})^*$, $f(d_k)$ is de£ned if $f(d_i)$ is de£ned for all $0 \leq i < k$ (here $d_0 = \epsilon$), and for all $i \leq k$, $\hat{\Sigma}_i \in f(d_{i-1})$, and $\tau_i \in M(\hat{\Sigma}_i) \cup \{\epsilon\} = \overline{M(\hat{\Sigma}_i)}$.

Remark 1: In order to implement a nondeterministic control policy a mechanism is needed for the on-line nondeterministic selection of the control action (from the set of choices computed off-line), and another mechanism is needed to determine when to nondeterministically change the control action. For the £rst purpose, a "coin toss" (with as many possible outcomes as the number of control action choices) can be used. For the second purpose, a "random timer" can be used. In the lack of any new observation, the control action is changed if and when the timer goes off.

Remark 2: Note that when f is a deterministic control policy, there is a unique choice for the control action selection at each occurrence of observation. Due to the uniqueness of choice, there is no need to maintain the control action selections made as part of the history. The history can simply be collapsed to the set Δ^* since given an observation history in Δ^* and a deterministic control policy f, it is possible to uniquely reconstruct the entire control & observation history in $(\Gamma \times \Delta)^*$.

In De£nition 1, while the selected control action $\hat{\Sigma}_i$ is in effect, the following possibilities exist for the events executed by the plant:

- Plant executes a sequence of enabled unobservable events followed by an enabled observable event, i.e., the plant executes a trace $t_i\sigma_i \in \Sigma^*\Sigma$ with $t_i \in (\hat{\Sigma}_i \cap M^{-1}(\epsilon))^*$ and $\sigma_i \in (\hat{\Sigma}_i - M^{-1}(\epsilon))$. In such a case, $M(\sigma_i) = \tau_i$.
- Plant executes a sequence of enabled unobservable events, and before the occurrence of any observable event the supervisor changes its control action, i.e., the plant executes a trace $t_i \in (\hat{\Sigma}_i \cap M^{-1}(\epsilon))^*$. Further in this case, $\tau_i = \epsilon$.

The above two cases can be combined into a single case by saying that while the selected control action $\hat{\Sigma}_i$ is in effect, and ith "observation" is τ_i, the plant executes a trace $t_i\sigma_i \in \Sigma^*\overline{\Sigma} = \Sigma^*$ such that $t_i \in (\hat{\Sigma}_i \cap M^{-1}(\epsilon))^*$ and $\sigma_i \in [\overline{\hat{\Sigma}_i} \cap M^{-1}(\tau_i)]$.

The above discussion leads to the following de£nition of traces enabled under a nondeterministic control policy.

De£nition 2: Consider a nondeterministic control policy $f : (\Gamma \times \overline{\Delta})^* \to 2^{\Gamma}$, and suppose $d_k = (\hat{\Sigma}_1, \tau_1) \cdots (\hat{\Sigma}_k, \tau_k) \in (\Gamma \times \overline{\Delta})^*$ is such that $f(d_k)$ is de£ned. Then the set of traces that are enabled by d_k is de£ned as:

$$\begin{aligned} Enabled(d_k) := \quad & \{s = (t_1\sigma_1 ... t_k\sigma_k) \in \Sigma^* \mid \\ & t_i \in (\hat{\Sigma}_i \cap M^{-1}(\epsilon))^*, \\ & \sigma_i \in [\overline{\hat{\Sigma}_i} \cap M^{-1}(\tau_i)]\} \end{aligned}$$

with $Enabled(d_0) = \{\epsilon\}$.

Remark 3: The above de£nition can recursively be stated as follows.

$$\begin{aligned} Enabled(d_0) = \quad & \{\epsilon\}; \\ Enabled(d_k) = \quad & Enabled(d_{k-1})\{t\sigma \mid \\ & t \in (\hat{\Sigma}_k \cap M^{-1}(\epsilon))^*, \\ & \sigma \in \overline{\hat{\Sigma}_k} \cap M^{-1}(\tau_k)\}. \end{aligned}$$

In the following we de£ne the closed-loop behavior achieved under a nondeterministic control policy.

De£nition 3: Let Σ be a set of events, Σ_u be the set of uncontrollable events, M be the observation mask, f be a non-deterministic control policy over (Σ_u, M), and (L, L_m) be the language model of a plant. Then the generated and the marked languages of the controlled plant under f, denoted by $(L/f, L_m/f)$, are de£ned as follows:

- $\forall s \in L$, $s \in L/f$ if and only if $\exists d_k \in (\Gamma \times \overline{\Delta})^*$ such that $s \in Enabled(d_k)$.
- $L_m/f := L/f \cap L_m$.

B. (Σ_u, M)-*Compatible Nondeterministic State Machine*

Having de£ned the notion of nondeterministic control policy we next de£ne the notion of control & observation compatible NSM, called (Σ_u, M)-compatible NSM. A (Σ_u, M)-compatible NSM restricts the behavior of a plant by operating in synchrony with the plant state machine. The motivation for de£ning (Σ_u, M)-compatible NSMs is to show that there exists a one-to-one correspondence between the class of nondeterministic control policies and the class of (Σ_u, M)-compatible NSMs in the sense that they enforce the same class of controlled behavior. This way a nondeterministic control policy can be more concisely represented as a (Σ_u, M)-compatible NSM. Through out this paper, it is assumed that for each state x of a NSM, there exists a silent transition (x, ϵ, x).

De£nition 4: Let $S = (X, \Sigma, \delta, X_0)$ be a nondeterministic state machine, and $\Sigma_u \subseteq \Sigma$ 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 $\forall x \in X$ and $\forall a \in \Sigma_u$, $\delta(x, a) \neq \emptyset$.
- S is called M-compatible if $\forall x \in X$ and $\forall a, b \in \overline{\Sigma}$, if $M(a) = M(b)$ and both $\delta(x, a)$ and $\delta(x, b)$ are nonempty, then $\delta(x, a) = \delta(x, b)$.
- S is called (Σ_u, M)-compatible if S is Σ_u-compatible and M-compatible.

Remark 4: The nondeterministic control exercised by a NSM $S = (X, \Sigma, \delta, X_0)$ is determined as follows. For $x \in X$, let $\Sigma_x := \{\sigma \in \Sigma \mid \delta(x, \sigma) \neq \emptyset\}$ be the set of events de£ned at x. From Σ_u-compatibility, $\Sigma_u \subseteq \Sigma_x$ for each $x \in X$, i.e., $\Sigma_x \in \Gamma$ for each $x \in X$.

The choice for the initial control action is given by the set, $\{\Sigma_x \subseteq \Sigma \mid x \in X_0\}$. The supervisor nondeterministically starts from one of its initial states, say $x_0 \in X_0$, thereby choosing (nondeterministically) it's initial control action to be Σ_{x_0}.

When at state x, depending on whether or not an ϵ-transition is de£ned at x, the following possibilities exist:

- Plant executes an enabled event $\sigma \in \Sigma_x$. Then the choice for the next control action is given by the set $\{\Sigma_y \subseteq \Sigma \mid y \in \delta(x, \sigma)\}$. The supervisor nondeterministically updates it's state to some $y_0 \in \delta(x, \sigma)$, thereby choosing (nondeterministically) it's next control action to be Σ_{y_0}.

  Note that the supervisor does not observe $\sigma \in \Sigma_x$ directly. However it's M-compatibility guarantees that the choice for the next control action is given by the same set for any $\sigma' \in M^{-1}M(\sigma)$.

- If an ϵ-transition is de£ned at x, then the supervisor can nondeterministically change it's control action by transitioning (nondeterministically) to one of the states in $\delta(x, \epsilon)$, say y_0. Then the new control action is given by Σ_{y_0}.

As in Remark 1, for a nondeterministic selection of initial state and a nondeterministic state update upon an observation, a "coin-toss" may be used; whereas for a nondeterministic control action change a "random-timer" may be used.

The following theorem establishes the one-to-one correspondence between the set of nondeterministic control policies and the set of (Σ_u, M)-compatible NSMs.

Theorem 1: The following relationship exists between nondeterministic control policies over (Σ_u, M) and (Σ_u, M)-compatible NSMs:

- For a nondeterministic control policy f over (Σ_u, M), there exists a (Σ_u, M)-compatible nondeterministic state machine S_f such that $L(S_f) = \Sigma^*/f$.
- For a (Σ_u, M)-compatible nondeterministic state machine S, there exists a nondeterministic control policy f_S over (Σ_u, M) such that $\Sigma^*/f_S = L(S)$.

III. M-RECOGNIZABILITY AND (Σ_u, M)-ACHIEVABILITY

In the previous section we showed that the class of controlled languages achieved by nondeterministic control policies and (Σ_u, M)-compatible NSMs are identical. In this section, we characterize this class of languages in terms of the property of achievability. We show that alike controllability both achievability and recognizability are preserved under union and intersection (in the later case, when restricted over pre£x-closed languages). We also present a polynomial algorithm for computing the in£mal pre£x-closed and achievable/recognizable superlanguage. This lets us verify achievability/recognizability polynomially.

De£nition 5: Let $K \subseteq \Sigma^*$, then

- K is said to be Σ_u-controllable with respect to Σ^* if $\forall s \in pr(K)$ and $\forall a \in \Sigma_u$, $sa \in pr(K)$.
- K is said to be M-recognizable with respect to Σ^* if $\forall s, t \in \Sigma^*$ and $\forall a \in \Sigma$ with $M(a) = \epsilon$, $sat \in pr(K)$ $\Rightarrow sa^*t \subseteq pr(K)$.
- K is said to be (Σ_u, M)-achievable with respect to Σ^* if K is Σ_u-controllable and M-recognizable with respect to Σ^*, and $\forall s, t \in \Sigma^*$, $\forall a \in \overline{\Sigma}, b \in \Sigma_u$ with $M(a) = M(b)$, $sat \in pr(K) \Rightarrow \{sbt\} \subseteq pr(K)$.

Theorem 2: (Σ_u, M)-achievability with respect to Σ^* is closed under set union over arbitrary languages and under set intersection over pre£x-closed languages.

For $K \subseteq \Sigma^*$, the class of pre£x-closed and (Σ_u, M)-achievable superlanguages of K with respect to Σ^* is de£ned as:

$$\overline{PA}_{\Sigma^*}(K) := \{K \subseteq K' \subseteq L \mid K' = pr(K'), K' \text{ is } (\Sigma_u, M)\text{-achievable wrt } \Sigma^*\}.$$

The in£mal pre£x-closed and (Σ_u, M)-achievable superlanguage of K with respect to Σ^* is denoted $inf\overline{PA}_{\Sigma^*}(K)$.

Lemma 1: $\forall K \subseteq \Sigma^*$, K is (Σ_u, M)-achievable with respect to Σ^* if and only if $pr(K) = inf\overline{PA}_{\Sigma^*}(K)$.

De£nition 6: Let $K \subseteq L = pr(L)$, then K is said to be (Σ_u, M)-achievable with respect to L if $pr(K) = inf\overline{PA}_{\Sigma^*}(K) \cap L$

Theorem 3: Let $K \subseteq L = pr(L)$, then K is (Σ_u, M)-achievable with respect to L if and only if there exists $K' \subseteq \Sigma^*$ such that K' is (Σ_u, M)-achievable with respect to Σ^* and $pr(K) = pr(K') \cap L$.

Theorem 4: (Σ_u, M)-achievability with respect to $L = pr(L) \subseteq \Sigma^*$ is closed under set union over arbitrary languages and under set intersection over pre£x-closed languages.

Next we develop an algorithm that shows a polynomial veri£cation of achievability is possible. We need the next result about the in£mal pre£x-closed and achievable superlanguage. For $K \subseteq L = pr(L)$, the class of pre£x-closed and (Σ_u, M)-achievable superlanguages of K contained in L is de£ned as:

$$\overline{PA}_L(K) := \{K \subseteq K' \subseteq L \mid K' = pr(K'), K' \text{ is } (\Sigma_u, M)\text{-achievable wrt } L\}.$$

The in£mal pre£x-closed and (Σ_u, M)-achievable superlanguage of K with respect to L is denoted $inf\overline{PA}_L(K)$.

Theorem 5: Let $K \subseteq L = pr(L)$. Then $inf\overline{PA}_L(K) = inf\overline{PA}_{\Sigma^*}(K) \cap L$.

We next present an algorithm for computing $inf\overline{PA}_L(K)$. From the result of the previous theorem we only need an algorithm for computing $inf\overline{PA}_{\Sigma^*}(K)$.

Algorithm 1: Let S_1 be a deterministic state machine such that $L(S_1) = pr(K)$, then we have the following algorithm for the computation of $inf\overline{PA}_{\Sigma^*}(K)$.

1) Separate the states in S_1: for every transition (x, b, y) with either $M(b) = \epsilon$ or $\exists(x, b', y')$ s.t. $M(b) = M(b') \neq \epsilon$, replace (x, b, y) by a pair of transitions (x, ϵ, x') and (x', b, y), where x' is a newly added state.
2) For every transition (x, b, y) with $M(b) = \epsilon$, add transitions (x, b, x) and (x, ϵ, y).
3) For every state x and every event $b \in \Sigma_u \cap M^{-1}(\epsilon)$, if b is not de£ned at x, then add b-labeled transitions to let $\delta(x, b) = \delta(x, \epsilon)$.
4) For every state x, every event $b \in \Sigma_u \cap (\Sigma - M^{-1}(\epsilon))$, and every transition (x, a, y) with $M(a) = M(b)$, add a transition (x, b, y).
5) For every state x and every event $b \in \Sigma_u \cap (\Sigma - M^{-1}(\epsilon))$, if no b-indistinguishable event a is de£ned at x, then add a transition $(x, b, dump)$, where $dump$ is an added state such that $\forall \sigma \in \Sigma_u$, $\delta(dump, \sigma) = \{dump\}$.
6) Return the modi£ed S_1 as the generator S_K of $inf\overline{PA}_{\Sigma^*}(K)$.

Theorem 6: Algorithm 1 is correct, i.e., $L(S_K) = inf\overline{PA}_{\Sigma^*}(K)$.

Remark 5: It is clear from the construction in Algorithm 1 that the complexity of computing $inf\overline{PA}_{\Sigma^*}(K)$ is linear in the number of states in the acceptor for K. Also, since $inf\overline{PA}_L(K) = inf\overline{PA}_{\Sigma^*}(K) \cap L$, the complexity of computing $inf\overline{PA}_L(K)$ is linear in the number of states of both the acceptor for K and the generator for L. In contrast in the deterministic setting, the complexity of computing the in£mal pre£x-closed controllable and observable superlanguage is exponential in the number of states of the acceptor for the language.

Further since K is (Σ_u, M)-achievable with respect to L if and only if $inf\overline{PA}_L(K) \subseteq pr(K)$, the complexity of verifying (Σ_u, M)-achievability of K with respect to L is quadratic in the number of states of the acceptor for K and linear in the number of states of the generator for L. This is the same as the complexity of verifying observability.

Theorem 7: Let $K \subseteq \Sigma^*$, then there exists a (Σ_u, M)-compatible non-deterministic state machine S_K such that $L(S_K) = pr(K)$ if and only if K is (Σ_u, M)-achievable with respect to Σ^*.

IV. Supervisory Control by Nondeterministic Policy

In this section we study supervisory control using nondeterministic control. The £rst theorem studies the "target" control problem and presents a necessary and suf£cient condition for the existence of a supervisor in terms of (Σ, M)-achievability. Later we study the "range" control problem.

Theorem 8: For $K \subseteq L = pr(L)$ there exists a nondeterministic control policy f over (Σ_u, M) such that $L/f = pr(K)$ if and only if K is (Σ_u, M)-achievable with respect to L.

Theorem 9: Given $A, E \subseteq L = pr(L)$, there exists a nondeterministic control policy f over (Σ_u, M) such that $A \subseteq L/f \subseteq E$ if and only if $inf\overline{PA}_L(A) \subseteq E$.

Remark 6: It follows from the two theorems of this section that both the target and range control problem using nondeterministic supervision is polynomially solvable. The target control problem requires the speci£cation to be achievable with respect to the plant, the complexity of veri£cation of which is quadratic in the number of states of the acceptor for the target speci£cation K, and linear in the number of states of the generator for the plant language L. The complexity of range control problem is linear in the number of states of all three: the acceptor for the lower bound speci£cation A, the acceptor for the upper bound speci£cation E, and the generator for the plant language L.

V. Conclusion

The supervisory control problem under partial observation using nondeterministic supervisors is formulated and studied in this paper. Various results are obtained.

There are three fold advantages of using nondeterministic controllers as opposed to deterministic ones: (i) Complexity reduction from being exponential to polynomial; (ii) Existence condition is weaker and so more speci£cations are attainable using nondeterministic supervisors, and (iii) Given a speci£cation, more sensors will be needed to satisfy observability than achievability (since achievability is a weaker requirement), and so there will be a saving in sensor cost when sensors for all events can be available.

References

[1] H. Cho and S. I. Marcus. Supremal and maximal sublanguages arising in supervisor synthesis problems with partial observations. *Mathematical Systems Theory*, 22:177–211, 1989.

[2] 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.

[3] N. B. Hadj-Alouane, S. Lafortune, and F. Lin. Centralized and distributed algorithm for on-line synthesis of maximal control policies under partial observation. *Discrete Event Dynamical Systems: Theory and Applications*, 6(41):379–427, 1996.

[4] K. Inan. Supervisory control: Theory and application to the gateway synthesis problem. In *Belgian-French-Netherlands Summer School on Discrete Event Systems*, page 25 pages. Spa, Belgium, 1993.

[5] 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.

[6] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, identi£cation and adaptive control*. Prentice Hall, 1986.

[7] R. Kumar and V. K. Garg. Optimal supervisory control of discrete event dynamical systems. *SIAM Journal of Control and Optimization*, 33(2):419–439, March 1995.

[8] R. Kumar and M. A. Shayman. Formulae relating controllability, observability, and co-observability. *Automatica*, 34(2):211–215, 1998.

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

[10] 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.

[11] K. Rudie and W. M. Wonham. The in£mal pre£x closed and observable superlanguage of a given language. *Systems and Control Letters*, 15(5):361–371, 1990.

[12] M. A. Shayman and R. Kumar. Process objects/masked composition: An object oriented approach for modeling and control of discrete event systems. *IEEE Transactions on Automatic Control*, 44(10):1864–1869, 1999.

[13] J. N. Tsitsiklis. On the control of discrete event dynamical systems. *Mathematics of Control Signals and Systems*, 2(2):95–107, 1989.