Reducing the Supervisory Control of Discrete-Event Systems under Partial Observation

Vahid Saeidi, Ali A. Afzalian, and Davood Gharavian

Abstract—Supervisor reduction procedure can be used to construct the reduced supervisor with a reduced number of states in discrete-event systems. However, it was proved that the reduced supervisor is control equivalent to the original supervisor with respect to the plant; it has not been guaranteed that the reduced supervisor and the original one are control equivalent under partial observation. In this paper, we extend the supervisor reduction procedure by considering partial observation; namely not all events are observable. A feasible supervisor which is constructed under partial observation becomes reduced based on control consistency of uncertainty sets of states, instead of the original supervisor. In order to construct a partial observation reduced supervisor, a partial observation control cover is constructed based on control consistency of uncertainty sets in the supervisor. Four basic functions are defined in order to capture the control and marking information on the uncertainty sets. In the resulting reduced supervisor, only observable events can cause state changes. The results are illustrated by some examples.

Index Terms—control consistency, control cover, discrete-event systems, partial observation, supervisor reduction.

I. INTRODUCTION

The state size and the computational complexity of a monolithic supervisor increase with state sizes of the plant and the specification [1], and may lead to state explosion [2]. However, the application of this theory is restricted, some works are reported on application of this theory in practice, e.g. [3, 4]. Although modular [5, 6] and incremental [7, 8] approaches try to overcome the complexity of the supervisor synthesis, other approaches tend to reduce a supervisor for simple implementation. The supervisor reduction procedure, given by [9], is an evolution of the proposed method in [10]. This procedure reduces the redundant information in the supervisor synthesis without any effect on controlled behavior. A reduced supervisor has some advantages comparing to the original supervisor, such as simplicity. Although this procedure is a heuristic method, it has been extended to other applications, e.g. coordination planning for distributed agents [11], supervisor localization procedure with full observation [12], and supervisor localization procedure under partial observation [13]. In [13], the authors employed the concept of relative observability to compute a partial-observation monolithic supervisor, and then they designed a localization procedure using (feasible) partial-observation supervisor to decompose the supervisor into a set of local controllers.

In this paper, we extend supervisor reduction procedure [9], to address the issue of partial observation. At first, we synthesize a partial-observation monolithic supervisor using the concept of relative observability [14]. Relative observability is stronger than observability [15, 16], weaker than normality [15, 16], and the supremal relative observable (and controllable) sublanguage of a given language exists. The supremal sublanguage may be effectively computed, and then implemented by a partial-observation (feasible and non-blocking) supervisor [13, 17]. Then, we suitably extend the supervisor reduction procedure in [9] to reduce a supervisor under partial observation.

In this paper, the partial-observation control cover is introduced. In particular, it is defined on the state set of the partial-observation supervisor; roughly speaking the latter corresponds to the power set of the full-observation supervisor's state set. As a result, a partial-observation reduced supervisor contains only observable state transitions.

The rest of the paper is organized as follows: In Section II, the necessary preliminaries are reviewed. Reducing the supervisory control under partial observation is proposed in Section III. In Section IV, five examples are given to clarify the proposed method. Finally, concluding remarks are given in Section V.

II. PRELIMINARIES

A discrete-event system (DES) is represented by an automaton $\mathrm{G} = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is a finite set of states, with $q_0 \in Q$ as the initial state and $Q_m \subseteq Q$ being the marked states; Σ is a finite set of events (σ) which is partitioned as a set of controllable events Σ_c and a set of uncontrollable events Σ_{uc}, where $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$. δ is a transition mapping $\delta: Q \times \Sigma \to Q$, $\delta(q, \sigma) = q'$ gives the next state q' is reached from q by the occurrence of σ. G is discrete-event model of the plant. In this context $\delta(q_0, s)!$ means that δ is defined for s at q_0. $L(\mathrm{G}) \coloneqq \{s \in \Sigma^* | \delta(q_0, s)!\}$ is the closed behavior of G and $L_m(\mathrm{G}) \coloneqq \{s \in L(\mathrm{G}) | \delta(q_0, s) \in Q_m\}$ is the marked behaviour of G [17, 18].

A set of all control patterns is denoted with $\Gamma = \{\gamma \in Pwr(\Sigma) | \gamma \supseteq \Sigma_{uc}\}$. A supervisory control for G is any map $V: L(\mathrm{G}) \to \Gamma$, where $V(s)$ represents the set of enabled events after the occurrence of the string $s \in L(\mathrm{G})$. The pair (G, V) is written V/G, to suggest " G under the supervision of V". A behavioral constraint on G is given by specification language $E \subseteq \Sigma^*$. Let $K \subseteq L_m(\mathrm{G}) \cap E$ be the supremal controllable sublanguage of E w.r.t. $L(\mathrm{G})$ and Σ_{uc}, i.e. $K = supC(L_m(\mathrm{G}) \cap E)$ [17]. If $K \neq \emptyset$, it can be shown as a DES, SUP =

Manuscript received 2 August 2017; accepted 9 October , 2017.

Vahid Saeidi (v_saeidi@sbu.ac.ir), Ali A. Afzalian (Afzalian@sbu.ac.ir), and D. Gharavian (d_gharavian@sbu.ac.ir) are with the department of Electrical Engineering, Abbaspour School of Engineering, Shahid Beheshti University, Tehran, Iran.

$(X, \Sigma, \xi, x_0, X_m)$, which is the recognizer for K. If G and E are finite-state DES, then K is regular language. Write $|.|$ for the state size of DES. Then $|\text{SUP}| \le |\text{G}||\text{E}|$. In applications, engineers want to employ RSUP, which has a fewer number of states (i.e. $|\text{RSUP}| \ll |\text{SUP}|$) and is control equivalent to SUP w.r.t. G [9], i.e.

$$L_m(\text{G}) \cap L_m(\text{RSUP}) = L_m(\text{SUP}), \quad (1)$$

$$L(\text{G}) \cap L(\text{RSUP}) = L(\text{SUP}). \quad (2)$$

The natural projection is a mapping $P: \Sigma^* \to \Sigma_0^*$ where (1) $P(\epsilon) := \epsilon$ (ϵ is the empty string), (2)for $s \in \Sigma^*$, $\sigma \in \Sigma$, $P(s\sigma) := P(s)P(\sigma)$, and (3) $P(\sigma) := \sigma$ if $\sigma \in \Sigma_0$ and $P(\sigma) := \epsilon$ if $\sigma \notin \Sigma_0$. The effect of an arbitrary natural projection P on a string $s \in \Sigma^*$ is to erase the events in s that do not belong to observable events set, Σ_0. The natural projection P can be extended and denoted with $P: Pwr(\Sigma^*) \to Pwr(\Sigma_0^*)$. For any $L \subseteq \Sigma^*$, $P(L) \coloneqq \{P(s) | s \in L\}$. The inverse image function of P is denoted with $P^{-1}: Pwr(\Sigma_0^*) \to Pwr(\Sigma^*)$ for any $L \subseteq \Sigma_0^*$, $P^{-1}(L) \coloneqq \{s \in \Sigma^* | P(s) \in L\}$. The synchronous product of languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ is defined by $L_1 \parallel L_2 = P_1^{-1}(L_1) \cap P_2^{-1}(L_2) \subseteq \Sigma^*$, where $P_i: \Sigma^* \to \Sigma_i^*$, $i = 1,2$ for the union $\Sigma = \Sigma_1 \cup \Sigma_2$ [19].

Let $\text{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be the recognizer of K, $\Sigma_0 \subseteq \Sigma$ and $P: \Sigma^* \to \Sigma_0^*$ be the natural projection. For $s \in \Sigma^*$, observation of $P(s)$ results in uncertainty as to the state of SUP given by the "uncertainty set" $U(s) \coloneqq \{\delta(q_0, s') | P(s') = P(s), s \in \Sigma^*\} \subseteq Q$. Uncertainty sets can be used to obtain a recognizer for the projected language $P(K)$. By definition of uncertainty set, each pair of states $x, x' \in X$, reachable by s, s', are control consistent, if there exists a non-blocking supervisor V such that $P(s') = P(s) \Longrightarrow V(s') = V(s)$. V is called a feasible supervisor [17]. Each pair of states $x, x' \in X$ in a monolithic supervisor can be considered one state in the feasible supervisor by self-looping an unobservable event σ, which occurs between states x, x'.

It was defined in [14], that K is relative observable w.r.t. $\bar{C}$, G and P (or $(\bar{C}, \text{G}, P)$-observable) for $K \subseteq C \subseteq L_m(\text{G})$, where $\bar{K}$ and $\bar{C}$ are prefix closed languages, if for every pair of strings $s, s' \in \Sigma^*$ such that $P(s) = P(s')$, the following two conditions hold,

$$(\forall \sigma \in \Sigma)\, s\sigma \in \bar{K}, s' \in \bar{C}, s'\sigma \in L(\text{G}) \Longrightarrow s'\sigma \in \bar{K}, \quad (4)$$

$$s \in K, s' \in \bar{C} \cap L_m(\text{G}) \Longrightarrow s' \in K. \quad (5)$$

In the special case, if $C = K$, then the relative observability property is tighten to the observability property. An observation property called normality was defined in [16], that is stronger than the relative observability. K is said to be normal w.r.t. $(L(\text{G}), P)$, if $P^{-1}P(\bar{K}) \cap L(\text{G}) = \bar{K}$, where $L(\text{G})$ is a prefix closed language and P is a natural projection.

III. REDUCING THE SUPERVISORY CONTROL UNDER PARTIAL OBSERVATION

Similar to the procedure, proposed in [9], to reduce the state size of the supervisory control with full observation, we propose a method to reduce the state size of the supervisory control under partial observation.

Let $\text{G} = (Q, \Sigma, \delta, q_0, Q_m)$, be the plant, $\Sigma_0 \subseteq \Sigma$ be the subset of observable events, and $P: \Sigma^* \to \Sigma_0^*$ be the corresponding natural projection. Also let $\text{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be the recognizer of supervisor K. Under partial observation, if $s \in L(\text{SUP})$ occurs, then $P(s)$ is observed. Let $U(s)$ be the subset of states that may be reached by some s'that looks like s, i.e.

$$U(s) = \{x \in X | (\exists s' \in \Sigma^*) P(s) = P(s'), x = \xi(x_0, s')\}.$$

Let $\mathcal{U}(X)$ be the set of uncertainty sets of all states in X, associated with strings in $L(\text{SUP})$, i.e.

$$\mathcal{U}(X) \coloneqq \{U(s) \subseteq X | s \in L(\text{SUP})\}$$

The transition function associated with $\mathcal{U}(X)$ is $\hat{\xi}: \mathcal{U}(X) \times \Sigma_0 \to \mathcal{U}(X)$. $\hat{\xi}$ is given by

$$\hat{\xi}(U, \sigma) = \bigcup \{\xi(x, u_1\sigma u_2) | x \in U, u_1, u_2 \in \Sigma_{u0}^*\}$$

Where $\Sigma_{u0} = \Sigma - \Sigma_0$. If there exist $u_1, u_2 \in \Sigma_{u0}^*$ such that $\xi(x, u_1\sigma u_2)!$ then $\hat{\xi}(U, \sigma)$is defined and denoted as $\hat{\xi}(U, \sigma)!$. Having $\mathcal{U}(X)$ and $\hat{\xi}$, partial observation monolithic supervisor SUPO can be defined. It is a feasible supervisor, and its synchronization by the plant is control equivalent to the original supervisor w.r.t. the plant. SUPO is defined as follows,

$$\text{SUPO} = (\mathcal{U}(X), \Sigma_0, \hat{\xi}, U_0, \mathcal{U}_m)$$

Where $U_0 = U(\epsilon)$ and $\mathcal{U}_m = \{U \in \mathcal{U}(X) | U \cap X_m \neq \emptyset\}$. It is known [13], that $L(\text{SUPO}) = P(L(\text{SUP}))$ and $L_m(\text{SUPO}) = P(L_m(\text{SUP}))$.

Let $U \in \mathcal{U}(X), x \in U$ be any state in SUP and $\alpha \in \Sigma_c$be a controllable event. We know that 1. α is enabled at $x \in U$, if $\xi(x, \alpha)!$, or 2. α is disabled at $x \in U, \neg\xi(x, \alpha)!$ and $(\exists s \in \Sigma^*)[\xi(x_0, s) = x \,\&\, \hat{\xi}(U_0, Ps) = U] \& \,\delta(q_0, s\sigma)!$ or 3. α is not defined at $x \in U$, if $\neg\xi(x, \alpha)!$ and $\neg\xi(x, \alpha)!$ and $(\exists s \in \Sigma^*)[\xi(x_0, s) = x \,\&\, \hat{\xi}(U_0, Ps) = U] \Longrightarrow \neg\delta(q_0, s\sigma)!$. Under partial observation, the control actions after string $s \in L(\text{SUP})$ depend on the uncertainty set $U(s) \in \mathcal{U}(X)$, i.e. the state of SUPO. It was proved that, if α is enabled at $x \in U$, then for all $x' \in U$, either α is also enabled at $x' \in U$, or α is not defined at $x' \in U$. On the other hand, if α is disabled at $x \in U$, then for all $x' \in U$, either α is also disabled at $x' \in U$, or α is not defined at $x' \in U$ [13].

In order to propose a supervisor reduction procedure under partial observation, consider the following four functions which capture the control and marking information on the uncertainty sets. Define $E: \mathcal{U}(X) \to Pwr(\Sigma_0)$ according to

$$E(U) = \{\sigma \in \Sigma_0 | (\exists x \in U) \xi(x, \sigma)!\}$$

$E(U)$ denotes the set of events enabled at state U. Also define $D: \mathcal{U}(X) \to Pwr(\Sigma_0)$ according to

$$D(U) = \left\{\sigma \in \Sigma_0 \,\middle|\, (\exists x \in U) \neg\xi(x, \sigma)! \,\&(\exists s \in \Sigma^*)[\xi(x_0, s) = x \,\&\, \delta(q_0, s\sigma)!]\right\}$$

$D(U)$ is the set of events, which are disabled at state U. Next, define $M: \mathcal{U}(X) \to \{0,1\}$ according to

$$M(U) = \begin{cases} 1, \text{if } (U \in U_m), \\ 0, \ \text{otherwise.} \end{cases}$$

$M(U) = 1$ if U is marked in SUPO, i.e. U contains a marked state of SUP. Finally define $T: \mathcal{U}(X) \to \{0,1\}$ according to

$$T(U) = \begin{cases} 1, & \text{if } (\exists s \in \Sigma^*)\xi(x_0, s) \in U \\ & \hat{\xi}(U_0, Ps) = U, \delta(q_0, s) \in Q_m \\ 0, & \text{otherwise.} \end{cases}$$

$T(U) = 1$ if U contains some states that correspond to a marked state of G, i.e. U contains a marked state of G. Now, the control consistency relation $\mathcal{R}_U \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ can be defined. $U, U' \in \mathcal{U}(X)$ are control consistent, i.e. $(U, U') \in \mathcal{R}_U$, if

$$E(U) \cap D(U') = E(U') \cap D(U) = \emptyset, \tag{6}$$

$$T(U) = T(U') \Rightarrow M(U) = M(U'). \tag{7}$$

Thus a pair of uncertainty sets (U, U') satisfies $(U, U') \in \mathcal{R}_U$, if (i) each event is enabled at least at one state of U, but is not disabled at any state of U', and vice versa; (ii) U, U' both contain marked states of SUP (both do not contain) provided that they both contain states corresponding to some marked states of G (both do not contain). It is easily verified that $\mathcal{R}_U$ is generally not transitive, thus it is not an equivalence relation. This leads to the partial-observation control cover. Let I be some index set, and $\mathcal{C}_U = \{\mathcal{U}_i \subseteq \mathcal{U}(X) | i \in I\}$ be a cover on $\mathcal{U}(X)$. $\mathcal{C}_U$ is a partial observation control cover, if

$$(i)(\forall i \in I)(\forall U, U' \in \mathcal{U}_i)(U, U') \in \mathcal{R}_U,$$

$$(ii)(\forall i \in I)(\forall \sigma \in \Sigma_0)(\exists U \in \mathcal{U}_i)\hat{\xi}(U, \sigma)! \Longrightarrow \left[(\exists j \in I)(\forall U' \in \mathcal{U}_i)\hat{\xi}(U', \sigma)! \Longrightarrow \hat{\xi}(U', \sigma) \in \mathcal{U}_j\right], \tag{8}$$

A partial observation control cover $\mathcal{C}_U$ lumps the uncertainty sets $U \in \mathcal{U}(X)$ into cells $\mathcal{U}_i \in \mathcal{C}_U, i \in I$ such that (i) the uncertainty sets U that reside in the same cell $\mathcal{U}_i$ must be pairwise control consistent, (ii) for every observable event $\sigma \in \Sigma_0$, the uncertainty set that is reached from any uncertainty set $U' \in \mathcal{U}_i$ by one-step transition σ, must be covered by the same cell $\mathcal{U}_j$. Obviously, two uncertainty sets U and U' belong to a common cell of $\mathcal{C}_U$, if and only if U and U' are control consistent, and two future uncertainty sets that can be reached respectively from U and U' by a given observable string are again control consistent. $\mathcal{C}_U$ is called a partial-observation control congruence if $\mathcal{C}_U$ happens to be a partition on $\mathcal{U}(X)$, namely its cells are pairwise disjoint. Having $\mathcal{C}_U$, $U_0 = U(\epsilon)$ and $x_0 \in U_0$, a generator $\mathrm{J} = (I, \Sigma_0, \zeta, i_0, I_m)$ can be defined over Σ_0 as follows,

$$\begin{aligned} & i_0 \in I \text{ such that } U_0 \in U_{i0}, \\ & I_m := \{i \in I | (\exists U \in \mathcal{U}_i) X_m \cap U \neq \emptyset\} \\ & \zeta: I \times \Sigma_0 \to I \text{ with } \zeta(i, \sigma) = j \\ & \text{if } (\exists U \in \mathcal{U}_i), \hat{\xi}(U, \sigma) \in \mathcal{U}_j; \end{aligned} \tag{9}$$

Note that, overlapping of some states results that i_0 and ζ may not be uniquely determined, and J may not be unique. If $\mathcal{C}_U$ is partition on $\mathcal{U}(X)$, J can be determined uniquely and it can be selected as the reduced supervisor, $\mathrm{RSUP_P}$.

We prove in Theorem 1, $\mathrm{RSUP_P}$ is control equivalent to SUP w.r.t. G.

Theorem 1: $\mathrm{RSUP_P}$ is control equivalent to SUP w.r.t. G, i.e.

$$L(\mathrm{G}) \cap L(\mathrm{RSUP_P}) = L(\mathrm{SUP}), \tag{10}$$

$$L_m(\mathrm{G}) \cap L_m(\mathrm{RSUP_P}) = L_m(\mathrm{SUP}). \tag{11}$$

Proof: We prove the claim in two steps, a. $\subseteq$, b. $\supseteq$.

a. As it was assumed that $L_m(\mathrm{SUP})$ is not empty, it follows that $L(\mathrm{G})$ and $L(\mathrm{RSUP_P})$ are not empty, and as they are closed, the empty string ϵ belongs to each. Now, suppose that $s \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$ implies that $s \in L(\mathrm{SUP})$ and $s\sigma \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$ such that $\sigma \in \Sigma$. We must prove that $s\sigma \in L(\mathrm{SUP})$. If $\sigma \in \Sigma - (\Sigma_c \cup \Sigma_0)$, then $s\sigma \in L(\mathrm{SUP})$, because $L(\mathrm{SUP})$ is controllable and observable. Now, assume $\sigma \in \Sigma_c \cap \Sigma_0$ and $s\sigma \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$. Since U and U' belong to the same cell $\mathcal{U}_i$, by definition of partial-observation control cover, they must be control consistent, i.e. $(U, U') \in \mathcal{R}_U$. Thus, $E(U) \cap D(U') = \emptyset$ which implies that $D(U') = \emptyset$. It means that all controllable and observable σ that is enabled at U, cannot be disabled at U'. Thus, $\forall x \in U'$, either $\xi(x, \sigma)!$ or $(\forall t \in \Sigma^*)[\xi(x_0, t) = x \;\&\; \neg\delta(q_0, t\sigma)!]$. Note that, $s\sigma \in L(\mathrm{G})$. Thus $\neg\delta(q_0, t\sigma)!$ is not true. Therefore, $\xi(x, \sigma)!$ is true, i.e. $s\sigma \in L(\mathrm{SUP})$.

Now, assume $s \in L_m(\mathrm{G}) \cap L_m(\mathrm{RSUP_P})$. It means that $\zeta(i_0, s) \in I_m$. From (9), it is obvious $\xi(x_0, s) \in X_m$, i.e. $s \in L_m(\mathrm{SUP})$.

b. Suppose that $s \in L(\mathrm{SUP})$ implies that $s \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$. Assume $s\sigma \in L(\mathrm{SUP})$. If $\sigma \in \Sigma - \Sigma_0$, then it is a self-loop transition at some states in $\mathrm{RSUP_P}$. thus, $s \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P}) \Rightarrow s\sigma \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$. If $\sigma \in \Sigma_0$, then (9) implies that $\zeta(i, \sigma) = j$. Thus, $s\sigma \in L(\mathrm{G}) \cap L(\mathrm{RSUP_P})$.

Now, assume $s \in L_m(\mathrm{SUP})$. It means that $\xi(x_0, s) \in X_m$. From (9), we can write $\zeta(i_0, s) \in I_m$. Namely, $s \in L_m(\mathrm{G}) \cap L_m(\mathrm{RSUP_P})$. The proof is complete.

Corollary 1: Let G be a non-blocking plant, described by closed and marked languages $L(\mathrm{G}), L_m(\mathrm{G}) \subseteq \Sigma^*$, and $\mathrm{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be the recognizer of the supervisor K, i.e. $K = L_m(\mathrm{SUP})$. Let $\mathrm{RSUP_P}$ be the reduced supervisor under partial observation. If K is relatively observable w.r.t. $(\bar{C}, \mathrm{G}, P)$, where $P: \Sigma^* \to \Sigma_0^*$, and $K \subseteq C \subseteq L_m(\mathrm{G})$, then $\mathrm{P(RSUP_P)}$ is control equivalent to P(SUP) w.r.t. G, i.e.

$$L_m(\mathrm{G}) \cap P^{-1}(L_m(\mathrm{P(RSUP_P)})) = L_m(\mathrm{G}) \cap P^{-1}(L_m(\mathrm{P(SUP)})),$$

$$L(\mathrm{G}) \cap P^{-1}(L(\mathrm{P(RSUP_P)})) = L(\mathrm{G}) \cap P^{-1}(L(\mathrm{P(SUP)})).$$

In order to clarify the proposed method for reducing a supervisor under partial observation, some examples are illustrated in the next section.

IV. EXAMPLES

In this section, we consider examples in order to verify the extended theory in Section III. The model construction and supervisor synthesis are carried out by TCT software [20]. A brief description of TCT procedures, which are used in this paper, is given in the Appendix.

Example 1: Let $\Sigma = \{1,2,3\}$ and G, SUP be the plant and the recognizer of supervisor, respectively (Fig. 1). Obviously, we can find C_1 and C_2 such that $K = L_m(\mathrm{SUP})$ is relatively observable w.r.t. $(\bar{C}_1, \mathrm{G}, P_1)$, where $P_1: \Sigma^* \to \Sigma_1^*$ and $\Sigma_1 = \{1,3\}$ and K is relatively observable w.r.t. $(\bar{C}_2, \mathrm{G}, P_2)$, where $P_2: \Sigma^* \to \Sigma_2^*$ and $\Sigma_2 = \{2,3\}$. But, we cannot find any C such that K is relatively observable w.r.t. $(\bar{C}, \mathrm{G}, P_0)$, where $P_0: \Sigma^* \to \Sigma_0^*$ and $\Sigma_0 = \{3\}$. We can find uncertainty sets $\mathcal{U}_1(X)$ and $\mathcal{U}_2(X)$ corresponding to P_1 and P_2, respectively.

a. G b. SUP

Fig. 1. The plant G and the corresponding supervisor, SUP

a. RSUP_1 b. RSUP_2

Fig. 2. Feasible reduced supervisors RSUP_1 and RSUP_2

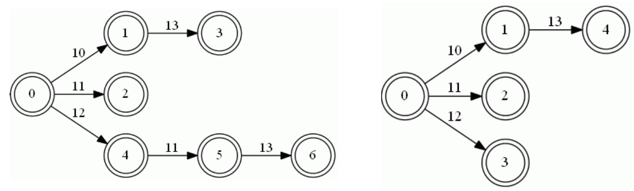
$\mathcal{U}_1(X) = \{\{0,1\},\{2\}\}$ and $\mathcal{U}_2(X) = \{\{0,2\},\{1\}\}$ can be constructed. The partial observation control covers can be constructed as $\mathcal{C}_1 = \mathcal{U}_1(X)$ and $\mathcal{C}_2 = \mathcal{U}_2(X)$. Thus, RSUP_1 and RSUP_2 are both feasible reduced supervisors, corresponding to P_1 and P_2, respectively (Fig. 2). Since states 1 and 2 are not control consistent, states 0, 1 and 2 in SUP cannot be lumped into one state, in order to construct a reduced supervisor. It is obvious that, RSUP_1 is control equivalent to SUP w.r.t. G, under natural projection P_1, but it is not control equivalent to SUP under P_2. Also, RSUP_2 is control equivalent to SUP w.r.t. G, under natural projection P_2, but it is not control equivalent to SUP under P_1 w.r.t. G.

Example 2: Let $\Sigma = \{10,11,12,13\}$ and G, SUP be the plant and the recognizer of supervisor, respectively (Fig. 3). We can find C_1, C_2 and C_3 such that $K = L_m(\mathrm{SUP})$ is relatively observable w.r.t. $(\bar{C}_1, \mathrm{G}, P_1)$, where $P_1: \Sigma^* \to \Sigma_1^*$ and $\Sigma_1 = \{10,12,13\}$ and is relatively observable w.r.t. $(\bar{C}_2, \mathrm{G}, P_2)$, where $P_2: \Sigma^* \to \Sigma_2^*$ and $\Sigma_2 = \{11,12,13\}$. Also, it is relatively observable w.r.t. $(\bar{C}_3, \mathrm{G}, P_3)$, where $P_3: \Sigma^* \to \Sigma_3^*$ and $\Sigma_3 = \{10,11,12\}$. Moreover, we can find C such that K is relatively observable w.r.t. $(\bar{C}, \mathrm{G}, P_0)$, where $P_0: \Sigma^* \to \Sigma_0^*$ and $\Sigma_0 = \{12\}$.

We can find the uncertainty set $\mathcal{U}(X) = \{\{0,1,2,4\},\{3\}\}$ corresponding to P_0. Note that RSUP_0 is the partial observation reduced supervisor, corresponding to control cover $\mathcal{C} = \{\{0,1,2,4\},\{3\}\}$ (Fig. 4). Since other control covers can be found corresponding to other uncertainty sets relevant to $P_i, i = 1,2,3$, the reduced supervisor is not unique. But, other feasible reduced supervisor seems have more number of states. Obviously, we can check that RSUP_0 is control equivalent to SUP under natural projection P_0 w.r.t. G.

Example 3: Supervisory control of transfer line under partial observation

Industrial transfer line consists of two machines M_1, M_2 and a test unit TU, which are linked by buffers B_1 and B_2 (Fig. 5). The capacities of B_1 and B_2 are assumed to be 3 and 1, respectively. If a work piece is accepted by TU, it is released from the system; if rejected, it is returned to B_1 for reprocessing by M_2. The specification is based on protecting B_1 and B_2 against underflow and overflow [17].



a. G b. SUP

Fig. 3. The plant G and the corresponding supervisor SUP

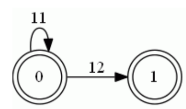


Fig. 4. The partial observation reduced supervisor RSUP_0

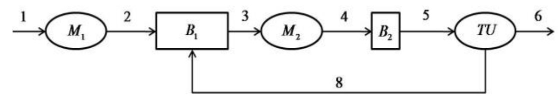
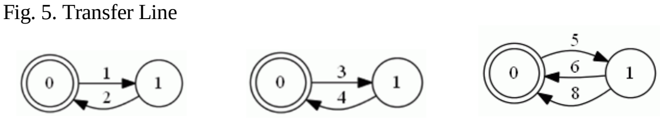
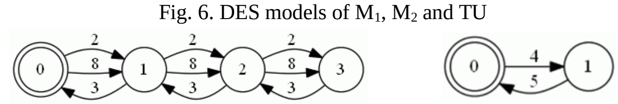


Fig. 5. Transfer Line



a. M_1 b. M_2 c. TU

Fig. 6. DES models of M_1, M_2 and TU



a. B_1 b. B_2

Fig. 7. Specifications of buffers

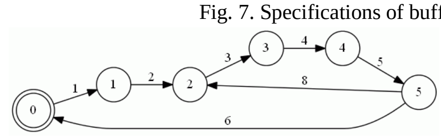


Fig. 8. The supremal relative observable supervisor for transfer line, (SUP)

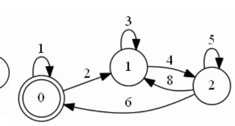


Fig. 9. $SUPO$ and $RSUP_0$

All events involved in the DES model are $\Sigma = \{1,2,3,4,5,6,8\}$, where controllable events are odd-numbered. State transition diagrams of M_1, M_2, TU and specifications of buffers are displayed in Figs. 6, 7, respectively. The recognizer of relative observable supervisor, SUP and the partial observation reduced supervisor, corresponding to $P_0: \Sigma^* \to \Sigma_0^*$, $\Sigma_0 = \Sigma - \{1,3,5\}$, are shown in Figs. 8, 9, respectively. We see that, events 1, 3, 5 appear just as self-loop transitions, each one at one state of the reduced supervisor, RSUP_0 (Fig. 9). Since the recognizer of partial observation supervisor, SUPO cannot be further reduced, RSUP_0 and SUPO are the same.

Example 4: Supervisory control of guide way under partial observation

Consider a guide way with two stations A and B, which are connected by a single one-way track from A to B on a guide way, as shown in Fig. 10. The track consists of 4 sections, with stoplights (*) and detectors (!) installed at various section junctions [17]. Two vehicles $\mathrm{V}_1, \mathrm{V}_2$ use the guide way simultaneously. $\mathrm{V_i}, \mathrm{i} = 1, 2$ may be in state 0 (at A), state j (while travelling in section $\mathrm{j} = 1, \dots, 4$), or state 5 (at B). The generator of V_i, $i = 1,2$ are shown in Fig. 11.

The plant to be controlled is $G = \mathrm{sync}(V_1, V_2)$. To prevent collision, control of the stoplights must ensure that V_1 and V_2

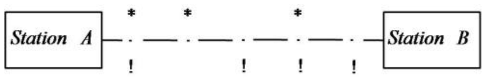


Fig. 10. Schematic of a guide way

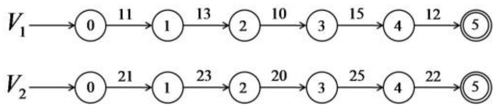


Fig. 11. DES model of each vehicle

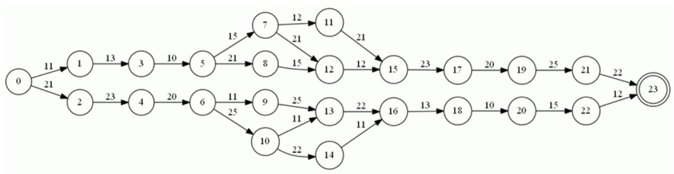


Fig. 12. The relative observable supervisor for the guide way $\Sigma_{uo} = \{13,23\}$, SUP

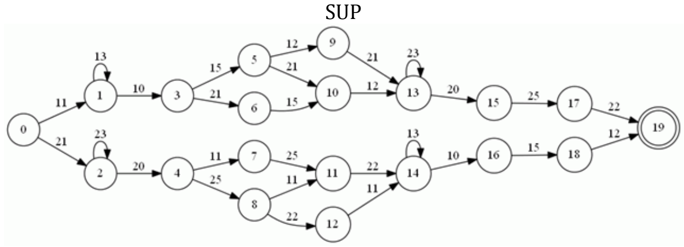


Fig. 13. The feasible supervisor for the guide way $\Sigma_{uo} = \{13,23\}$, SUPO

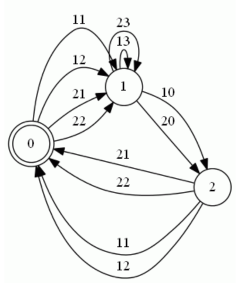


Fig. 14. The partial observation reduced supervisor for the guide way, RSUP_0

never travel on the same section of track simultaneously. Namely, V_i, $i = 1,2$ are mutual exclusion of the state pairs (i, i), $i = 1,..,4$. Controllable events are odd-numbered and the unobservable events 13, 23 are considered to synthesize the supremal relative observable supervisor, i.e. $P_0: \Sigma^* \rightarrow \Sigma_0^*,\ \Sigma_0 = \Sigma - \{13, 23\}$. The supremal relative observable supervisor, SUP is shown in Fig. 12, and its corresponding partial observation supervisor SUPO is shown in Fig. 13. The reduced supervisor, in which unobservable events 13, 23 are self-looped at state 1, is shown in Fig. 14.

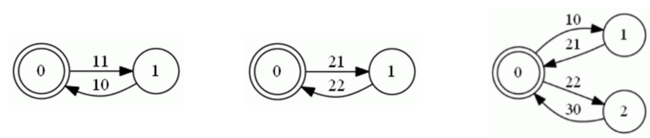
Moreover, events 15, 25 are self-looped at all states of the reduced supervisor (hence, they are not shown). Thus, the supervisor is normal w.r.t. $(L_m(G), P_N)$, where $P_N: \Sigma^* \rightarrow \Sigma_N^*, \Sigma_N = \Sigma - \{15, 25\}$. It can be checked that $P_0(\mathrm{RSUP}_0)$ and $P_0(\mathrm{SUP})$ are isomorph. Moreover, if the supervisor does not observe events 13, 23, they cannot be disabled at states 0, 2 in RSUP_0. It means that, they appear as self-loop transitions at states 0, 2. But the state size of the reduced supervisor does not change.

Example 5: Supervisory control of AGV under partial observation

A work cell consists of two machines M_1, M_2 and an automated guided vehicle AGV as shown in Fig. 15. AGV can be loaded with a work piece either from M_1 (event 10) or from M_2 (event 22), which it transfers respectively to M_2 (event 21) or to an output conveyor (event 30) [17]. Let CELL=sync(M_1,M_2,AGV). We can see CELL is blocking in state 9, i.e. the sequence of events reaches to a state from which no further transitions are possible (Fig. 16). To prevent blocking, we define SPEC=trim(CELL), as an appropriate specification (Fig. 17). The supremal relative observable supervisor, SUP is shown in Fig. 18, and its corresponding partial observation supervisor SUPO is shown in Fig. 19. In Fig. 19, states 0, 3 and states 1, 2 are control consistent, respectively. Thus, the partial observation based reduced supervisor, RSUP_0 is as shown in Fig. 20. Assume $P_0: \Sigma^* \rightarrow \Sigma_0^*,\ \Sigma_0 = \Sigma - \{11\}$, we can easily check that Corollary 1 is satisfied for SUP and RSUP_0.

V. Conclusions

This paper addresses an extension to supervisor reduction procedure, proposed in [9], by considering partial observation; namely not all events are observable. We reduced a feasible partial observation supervisor instead of the original one. In the resulting reduced supervisor, only observable events can cause state changes. We finally clarified the extended theory by some examples.



a. M_1 b. M_2 c. AGV

Fig. 15. DES model of each machine and AGV

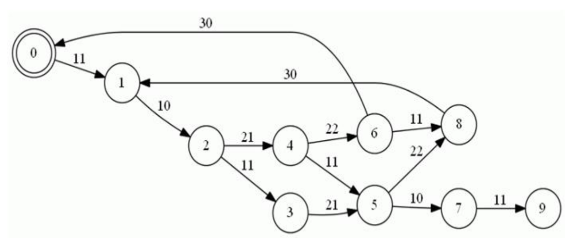


Fig. 16. DES model of CELL

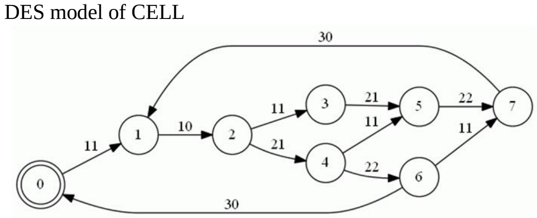


Fig. 17. DES model of specification, SPEC

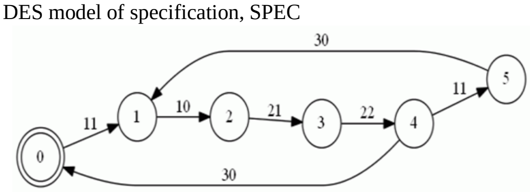


Fig. 18. The relative observable supervisor of AGV $\Sigma_{uo} = \{11\}$, SUP

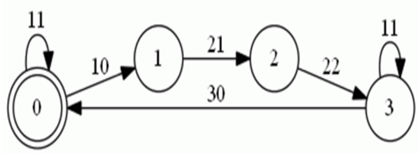


Fig. 19. The feasible supervisor of AGV $\Sigma_{uo} = \{11\}$, SUPO

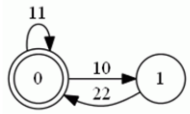


Fig. 20. The partial observation reduced supervisor for AGV, RSUP_0

REFERENCES

[1] P. Gohari, W. M. Wonham, On the complexity of supervisory control design in the RW framework, IEEE Trans. Syst. Man Cybern., Special Issue DES, vol.30, pp.643–652, 2000.

[2] W. M. Wonham, P. J. Ramadge, Modular supervisory control of discrete-event systems, Math. Control Signal Syst., vol.1, pp.13-30, 1988.

[3] R. J. M. Theunissen, M. Petreczky, R. R. H. Schiffelers, D. A. van Beek, J. E. Rooda, Application of Supervisory Control Synthesis to a Patient Support Table of a Magnetic Resonance Imaging Scanner, IEEE Trans. Autom. Sci. Eng., vol.11, pp.20-32, 2014.

[4] A. Afzalian, A. Saadatpoor, W. M. Wonham, Systematic supervisory control solutions for under-load tap-changing transformers, Control Engineering Practice, vol.16, pp.1035-1054, 2008.

[5] Y. Willner, M. Heymann, Supervisory control of concurrent discrete-event systems, International Journal of Control, vol.54, pp.1143-1169, 1991.

[6] J. Komenda, J. H. Van Schuppen, Modular Control of Discrete-Event Systems with Coalgebra, IEEE Trans. Autom. Control, vol.53, pp.447-460, 2008.

[7] H. Marchand, S. Pinchinat, Supervisory control problem using symbolic bisimulation techniques, in Proc. of Amer. Control Conf., Chicago, IL, pp.4067–4071, 2000.

[8] A. Vahidi, M. Fabian, B. Lennartson, Efficient supervisory synthesis of large systems, Control Engineering Practice, vol.14,pp.1157-1167, 2006.

[9] R. Su, W. M. Wonham, Supervisor reduction for discrete-event systems, Discrete-event Dyn. Syst., vol.14, pp.31–53, 2004.

[10] A. F. Vaz, W. M. Wonham, On supervisor reduction in discrete-event systems, International J. Control, vol.44, pp.475-491, 1986.

[11] K. T. Seow, M. T. Pham, C. Ma, M. Yokoo, Coordination Planning: Applying Control Synthesis Methods for a Class of Distributed Agents, IEEE Trans. Control Syst. Technol., vol.17, pp.405-415, 2009.

[12] K. Cai, W. M. Wonham, Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems, IEEE Trans. Autom. Control, vol.55, pp.605-618, 2010.

[13] R. Zhang, K. Cai, W. M. Wonham, Supervisor localization of discrete-event system s under partial observation, Automatica, vol.81, pp.142-147, 2017.

[14] K. Cai, R. Zhang, W.M. Wonham, Relative Observability of Discrete-Event Systems and its Supremal Sublanguages, IEEE Trans. Autom. Control, vol.60, pp.659- 670, 2015.

[15] F. Lin, W. M. Wonham, Onobservability of discrete-event systems, Information Sciences, vol.44, pp.173-198, 1988.

[16] R. Cieslak, C. Desclaux, A. S. Fawaz, P. Varaiya, Supervisory control of discrete-event processes with partial observations, IEEE Trans. Autom. Control, vol.33, pp.249-260, 1988.

[17] W. M. Wonham, Supervisory control of discrete-event systems, Lecture Notes University of Toronto, 2016, http://www.control.utoronto.ca/DES.

[18] C.G. Cassandras, S. Lafortune, Introduction to Discrete-event Systems, 2nd edn. Springer, NewYork, 2008.

[19] L. Feng, W. M. Wonham, Supervisory Control Architecture for Discrete-Event Systems, IEEE Trans. Autom. Control, vol.53, pp.1449-1461, 2008.

[20] W. M. Wonham, Control Design Software: TCT. Developed by Systems Control Group, University of Toronto, Canada, 2014, http://www.control.utoronto.ca/cgi-bin/dlxptct.cgi.

Appendix

In this appendix, a quick review of TCT commands is presented.

DES=sync(DES1,DES2,...,DESk) is the synchronous product of DES1,DES2,...,DESk.

DES3= supcon(DES1, DES2) for a controlled generator DES1, forms a trim recognizer for the supremal controllable sublanguage of the marked (“legal”) language generated by DES2 to create DES3. This structure provides a proper supervisor for DES1.

DES3=supconrobs(DES1,DES2,[NULL/IMAGE/IMAGE_DES]) is a trim DES which represents the supremal controllable and relatively observable sublanguage of the legal language represented by DES2, with respect to the plant DES1 and natural projection specified by the listed Null or Image events, or the latter’s allevents representation.

DAT3= condat(DES1, DES2) returns control data DAT3 for the supervisor DES2 of the controlled system DES1. If DES2 represents a controllable language (with respect to DES1),as when DES2 has been previously computed with supcon, then condat will display the events that are disabled at each state of DES2. In general, condat can be used to test whether a given language DES2 is controllable: just check that the disabled events tabled by condatare themselves controllable (have odd-numbered labels).

DES3=supreduce(DES1, DES2, DAT2) is a reduced supervisor for plant DES1 which is control-equivalent to DES2, where DES2 and control data DAT2 were previously computed using supcon and condat. Also returned is an estimated lower bound slb for the state size of a strictly state-minimal reduced supervisor. DES3 is strictly minimal if its reported state size happens to equal the slb.

DES2=project(DES1, NULL/IMAGE EVENTS) is a generator of the projected closed and marked languages of DES1, under the natural projection specified by the listed Null or Image events.

True/False= isomorph(DES1, DES2) tests whether DES1 and DES2 are identical up to renumbering of states; if so, their state correspondence is displayed.