2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Hierarchical Robust Supervisory Control of Discrete-Event Systems

Mohsen Zamani Fekri† and Shahin Hashtrudi-Zad‡

Abstract— **In this paper, the problem of robust supervisory control of a finite family of discrete-event plants is studied. Each plant has a separate closed specification language. A hierarchical solution which can be regarded as an extension of Zhong and Wonham approach to hierarchical control, is developed. It is shown that suitable extensions of Output-Control-Consistency and Strict-Output-Control-Consistency properties can be used to establish a high-level control structure and to ensure hierarchical consistency.**

I. Introduction

Ramadge and Wonham [10] were first to introduce the supervisory control in discrete-event systems (DES). One of the most active areas in the research so far, has been the Robust Supervisory Control (RSC) of DES which was initiated by Lin [8]. Research was continued on this problem in different frameworks, e.g. [1], [12], [4]. [1], [12] present a complete solution for the case in which the number of the plants models are finite and each model has its own specification. One of the important applications of robust supervisory control is in fault recovery problems [11], [12]. In order to deal with computational complexity of designing DES supervisors and to have more transparency in design, researchers have explored modular [14] and hierarchical [15], [2], [3], [7] methodologies. Specifically Zhong and Wonham [15] initiated a bottom-up hierarchical approach in which the high-level is obtained from abstracting the low-level. Consistency in exchanging the information between the two layers is established in [15] by refining the information sent up to the high-level. In [9] authors have explained a hierarchical system with uncertainty consisting of several models at the low-level and one specification at the high-level.

In this paper we have extended the RSC problem of [1] and [12] to the hierarchical framework of [15] for the case of full observation and closed languages. These assumptions are not necessary and relaxing them is the subject of our ongoing research. Our main results include a hierarchical solution for the problem of robust supervisory control of a finite family of DES plants in which each plant has a separate closed specification language. It is shown that suitable extensions of Output-Control-Consistency (OCC) and Strict-Output-Control-Consistency (SOCC) properties can be used to establish a high-level control structure and to ensure hierarchical consistency. The main difference between this work and [9] is that here each plant has its own specification. furthrmore here we introduce and discuss joint-OCC and joint-SOCC properties and show how they can guarantee hierarchcal consistency.

The paper is organized as follows. Section II reviews the RSC problem of [1] and [12] and Hierarchical Supervisory control (HSC) problem of [15]. Section III explains the developments of high-level model. Section IV presents the main results of the paper and discusses the low-level implementation of the supervisor, the relation between the high and the low-level systems under supervision and the hierarchical consistency are discussed. The proofs have been removed because of space limitation.

II. Preliminaries

A. Supervisory Control

The supervisory control problem addressed in this paper, has been posed in the Ramadge-Wonham (RW) supervisory framework [10]. An automaton G is described here by a 5-tuple $G = (Q, \Sigma, q_0, \delta, Q_m)$ where Q, Σ and Q_m denote the set of states, the set of the events and the set of marked states respectively, q_0 is the initial state and δ is the partial transition function. In this work we assume the full marking case and hence we have $Q_m = Q$. $L(G)$ denotes the closed language of the automaton G. Supervision requires us to partition the set of events Σ into controllable and uncontrollable subsets. Here $\Sigma = \Sigma_c \cup \Sigma_{uc}$ where Σ_c and Σ_{uc} denote the controllable and uncontrollable subsets respectively. Let $E \subseteq L(G)$ denote the legal language (i.e. the string in L(G) that satisfy design specifications). We call a language K controllable with respect to another closed language L if $\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}$. Given a plant model G and a specification E, a controller $K \subseteq E$ is a maximal subset of E which is controllable with respect to $L(G)$ and is implementable through running a supervisor S. Structurally, a supervisor is a map $S : \Sigma^* \to \Gamma$ where $\Gamma = \{\gamma \subseteq \Sigma | \gamma \supseteq \Sigma_{uc}\}$ and $S(s)$ is the subset of events which are enabled after the execution of $s \in \Sigma^*$. The language of the system under supervision is referred to as $L(S/G)$.

B. Robust Supervisory Control

When the system is represented by several models (for example, due to the dynamics uncertainty or occurrence of the faults) the robust supervisory control methodologies come in useful. The robust supervisory control framework we use in this paper was originally proposed by Lin [8] and later was extended in [1], [12]. In this approach, the number of the plant models is finite and each model G_k has its own specification E_k. it is assumed the plant models G_k agree on the controllability of the events. The version of robust supervisory control problem which we tend to extend to hierarchical framework is given in Theorem 2.1:

†, ‡: Both authors are with the ECE Department of Concordia University, Montreal, Canada. Emails: mohse_za, shz@ece.concordia.ca

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

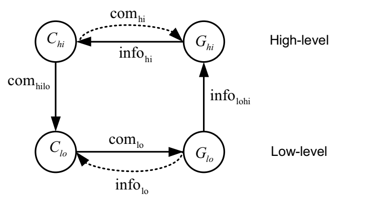


Fig. 1. General overview of a hierarchical supervisory control system

Robust Supervisory Control Problem: Consider a finite set of plants G_k and a set of legal languages $E_k \subseteq L(G_k)$ for $1 \leq k \leq n$. A supervisor $S : \Sigma^* \to \Gamma$ where $\Sigma = \bigcup\limits_{1 \leq k \leq n} \Sigma_k$ and $\Gamma = \{\gamma \subseteq \Sigma | \gamma \supseteq \bigcup\limits_{1 \leq k \leq n} \Sigma_{uc,k}\}$ is said to be robust if we have $L(S/G_k) \subseteq E_k$.

Theorem 2.1: [1], [12] Let G be any automaton whose closed behavior is $L(G) = \bigcup\limits_k L(G_k)$. A robust supervisor S exists that solves the RSC problem if and only if there exists a language $K \neq \emptyset$, $K \subseteq \bigcap\limits_k (E_k \cup [\Sigma^* - L(G_k)]) \cap L(G)$ such that it is controllable w.r.t $L(G)$. Furthermore $L(S/G) = K$ and for a model G_k we have $L(S/G_k) = K \cap L(G_k)$. ■

C. Hierarchical Supervisory Control

The hierarchical supervisory control (HSC) problem setup presented in [15], [5], [6] takes advantage of the reduced-size system at the high-level to design the supervisor and provide a more transparent design. A review of the hierarchical setup proposed in [15] follows. Fig. 1 shows the hierarchical structure that is implemented virtually during the running time. In this configuration G_{lo} is a finite-state deterministic automaton which represents the actual plant model, controlled in real world, and G_{hi} is the finite state deterministic automaton which represents the abstracted information from the the plant model G_{lo}. For a detailed discussion on this configuration the reader can refer to [15]. The channel info$_{lohi}$ is a causal map $\theta : L(G_{lo}) \longmapsto T^*$ where T is the set of events at the high-level. The causal map θ would have the following property:

$$\theta(\varepsilon) = \varepsilon$$

$$\theta(s\sigma) = \begin{cases} \text{either} & \theta(s) \\ \text{or} & \theta(s)\tau, \quad \text{for some } \tau \in T \end{cases} \tag{1}$$

An output map $\omega : L(G_{lo}) \longmapsto T_o$ where $T_o = T \cup \{\tau_o\}$ and τ_o is called the silent event, is defined over $L(G)$ so that for every $s \in L(G_{lo})$

$$\omega(s\sigma) = \begin{cases} \tau_o & \text{if} \quad \theta(s\sigma) = \theta(s) \\ \tau & \text{if} \quad \theta(s\sigma) = \theta(s)\tau. \end{cases} \tag{2}$$

So the output map ω generates the silent event τ_o whenever the map θ outputs nothing and generates the new high-level event τ otherwise.

The output map ω and the automaton G_{lo} can be combined into a Moore automaton $G_{lo} = (Q, \Sigma, T_0, \delta, \omega, q_0, Q)$ as elaborated in [15]. Now let $L_{lo} = L(G_{lo})$ and $L_{hi} = \theta(L_{lo})$, i.e. L_{hi} is the abstracted language at the high-level. Also let $G_{hi} = (Q_{hi}, T, q_{0,h}, \delta_{hi}, Q_{hi})$ be an automaton (at the high-level) with the event set T whose closed behavior generates L_{hi}, i.e. we have $L_{hi} = L(G_{hi})$. To equip the high-level model G_{hi} with control structure, we need to know how the controllability is inherited at the high-level. To do so, the vocal states are defined. A state is called vocal if the path s ending in that state generates a non-silent output, i.e. $\omega(s) \neq \tau_o$. A path (string) connecting two vocal states to each other or the root state to a vocal state, is called a *silent path* if all of its intermediate states are silent. A silent path is called controllable if at least one of its events is controllable and is called uncontrollable if all of its events are uncontrollable. Every silent path corresponds to a high-level event and furthermore the high-level event associated with a silent path is called controllable if that silent path is controllable. Uncontrollable events at the high-level are defined similarly. A silent path is colored red if it is controllable and is colored green if it is uncontrollable. Once a vocal state in G_{lo} is reached a high-level event is generated. So from the controllability point of view it should be clear whether a vocal state has been reached through red or green paths. It might be the case that a vocal state is reachable by both red and green paths.

Definition 2.2: [15] A Moore automaton G_{lo}, as previously defined, is said to satisfy the Output-Control-Consistency (OCC) property if every vocal state in it, is reachable either by only red silent paths or by only green silent paths. ■

Consider a partition of the high-level event set $T = T_c \cup T_{uc}$ where T_c and T_{uc} denote the controllable and uncontrollable subset of T respectively. Also let $L_{voc} = \{s | s \in L(G_{lo})$ and $\omega(s) \neq \tau_0\}$ be the set of strings with vocal ends. Now for $s \in L_{voc}$ define $\lambda : L_{voc} \longmapsto \{red, green\}$ to be:

$$\lambda(s) = \begin{cases} red & \omega(s) \in T_c \\ green & \omega(s) \in T_{uc} \end{cases} \tag{3}$$

and $color : L_{voc} \longmapsto \{red, green\}$ to be:

$$color(s) = \begin{cases} red & \text{final silent segment of s is red} \\ green & \text{final silent segment of s is green} \end{cases} \tag{4}$$

Therefore we have the following paraphrasing of definition 2.2:

Definition 2.3: Let G_{lo} be a Moore automaton with the following structure:

$$G_{lo} = (Q, \Sigma, T_o, \delta, \omega, q_0, Q) \tag{5}$$

. G_{lo} is said to satisfy the OCC property if for each $s \in L_{voc}$, we have:

$$\lambda(s) = red(green) \Longleftrightarrow color(s) = red(green). \tag{6}$$

■

If the system satisfies the OCC condition, a control structure C_{hi} can be computed for the high-level by a control law γ_{hi} which is implemented through a *disabled-event map* Δ_{hi} :

$L(G_{hi}) \longmapsto pwr(T_c)$. Δ_{hi} denotes the set of events which are disabled after $t \in L_{hi}$. A low-level version of *disabled-event map* Δ_{hi} is defined by $\Delta_{lo} : \Sigma^* \times T \rightarrow pwr(\Sigma_c)$ as follows:

$$\begin{aligned} \Delta_{lo}(s,t) \quad &= \quad \{\sigma \in \Sigma_{uc} | \; (\exists s' \in \Sigma_u^*) \; s\sigma s' \in L_{lo} \; \& \; \omega(s\sigma s') \\ & \quad\quad \in \Delta_{hi}(t) \; \& \; \forall s'' < s', \; \omega(s\sigma s'') = \tau_o\} \end{aligned} \qquad (7)$$

The control at the low-level is implemented through a version of γ_{hi}, say $\gamma_{lo} : \Sigma^* \times \sigma \longmapsto \{0,1\}$.

Let $E_{hi} \subseteq L_{hi}$ be a controllable legal language at the high-level. E_{hi} can be synthesized as the controlled behavior of G_{hi} under the supervision of γ_{hi}, i.e. $L(\gamma_{hi}/G_{hi}) = E_{hi}$. There would be a corresponding control law γ_{lo} at the low-level that translates the high-level commands to the low-level. Also let $E_{lo} = \theta^{-1}(E_{hi}) \subseteq L_{lo}$ be the low-level legal language correspondent to E_{hi}. Theorem 2.4 states the relation between the control low γ_{lo} and the legal language E_{lo} at the low-level:

Theorem 2.4: [15] For the above setup:

$$L(\gamma_{lo}/G_{lo}) = E_{lo}^{\uparrow} \qquad (8)$$

where $E_{lo}^{\uparrow}$ is the supremal controllable sublanguage of E_{lo} w.r.t L_{lo} and Σ_{uc}. ■

It is desired to have $\theta(L(\gamma_{lo}/G_{lo})) = E_{hi}$; in other words, E_{hi} is to be recovered through the low-level implementation. Strictly-Output-Control-Consistency can gaurantee the above property.

Definition 2.5: [15], [6] Two red vocal states n_1 and n_2, are said to be partners if their silent paths start either at the root state or at the same vocal state, they share an initial segment labeled $s_1\sigma$ with $\sigma \in \Sigma_c$ and this shared segment is followed in turn by segments labeled by string $s_2 s_3$ and $s_2 s_4$ respectively, where $s_2 \in \Sigma_{uc}^*$ and at least one of the strings s_3 and s_4 belongs to Σ_{uc}^*. ■

Definition 2.6: [15] A plant model G_{lo} is said to be Strictly-Output-Control-Consistent (SOCC) if i) it is OCC and ii) no two red vocal states are partners in it. ■

If G_{lo} satisfies the SOCC property, the image of $L(\gamma_{lo}/G_{lo})$ under the map θ would be the original high-level controllable language E_{hi}. Theorem 2.7 summarizes the result.

Theorem 2.7: [15] Assume that the low-level model G_{lo} satisfies the SOCC property. Let θ, E_{hi}, E_{lo} and γ_{lo} be as defined previously. Then we have:

$$\theta(L(\gamma_{lo}/G_{lo})) = E_{hi}. \qquad (9)$$

■

We say *hierarchical consistency* holds in the system if $\theta(L(\gamma_{lo}/G_{lo})) = E_{hi}$.

III. Proposed Control Structure

We aim to solve a RSC problem for a finite number of models, representing different aspects of a plant dynamics, in a hierarchical framework. Fig. 2 shows a Hierarchical robust Supervisory Control (HRSC) configuration for a system consisting of two models. A detailed definition of HRSC problem is given below. Consider a set of low-level Moore automata $G_{l,k} = (Q_k, \Sigma_k, T_o, \delta_k, \omega_k, q_{0,k}, Q_k)$, $(k = 1,..,n)$. We assume all the models agree on the controllability of the events. Our objective is to design a robust supervisor for the above plant models. We want to develop a hierarchical solution for the robust control problem. So we assume there exists a map $\theta : \bigcup_k L(G_{l,k}) \longmapsto T^*$ that reports the important sequences from the low to the high-level. We will discuss in section III.A how θ can be obtained. Following the approach in [15], let $G_{h,k}$ denote the abstracted model at the high-level that generates $\theta(L(G_{l,k}))$. We also assume the desired behavior is given in the form of closed languages $E'_{h,k}$ for each $G_{h,k}$ at the high-level. With the legal behavior $E'_{h,k}$ given at the high-level, we build the equivalent low-level legal languages $E'_{l,k} = \theta^{-1}(E'_{h,k})$. With the previous notations, a road map of the solution follows.

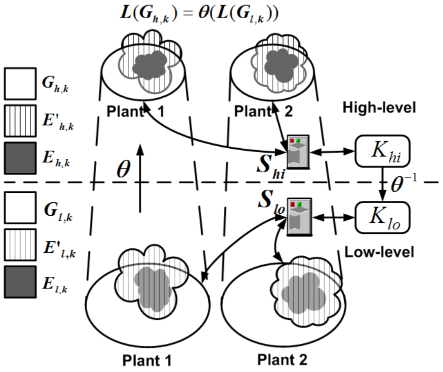


Fig. 2. HRSC problem configuration with two models

The RSC problem is solved at the high-level for the models $G_{h,k}$ and the specifications $E'_{h,k}$, yielding a robust supervisor S_{hi}. The systems under the supervision at the high-level would be $L(S_{hi}/G_{h,k}) = E_{h,k} \subseteq E'_{h,k}$. Let S_{lo} be the implementation of S_{hi} at the low-level which will be discussed in details later. The relation between the system under the supervision at the high-level, i.e. $E_{h,k} = L(S_{hi}/G_{h,k})$ and that under the supervision at the low-level, i.e. $L(S_{lo}/G_{l,k})$ needs to be investigated.

A. Information mapping

In order to transmit the low-level sequences of the events to the high-level and also to translate the high-level commands for the low-level, we need to have a map capable of uniquely transmitting the low-level strings to the high-level. In the reverse direction, the map would not be a function and a high-level sequence might be translated to several low-level sequences.

Consider a set of low-level Moore automata $G_{l,k} = (Q_k, \Sigma_k, T_o, \delta_k, \omega_k, q_{0,k}, Q_k)$, $(k = 1,..,n)$ where $\omega_k : L(G_{l,k}) \longmapsto T_o$ are the individual output maps, defined similarly as in (2), and T_o is the high-level event set. Each output map ω_k in fact corresponds to a map $\theta_k : L(G_{l,k}) \longmapsto T^*$. The image of the language $L(G_{l,k})$ under the map θ_k would be a language $L_{h,k} \subseteq T^*$ $(L_{h,k} = \theta_k(L(G_{l,k})))$. Let $G_{h,k}$ be any automaton whose closed behavior is $L_{h,k}$, i.e. we have:

$$L(G_{h,k}) = \theta_k(L(G_{l,k})) \tag{10}$$

All the high-level languages $L(G_{h,k})$ $(k=1,..,n)$, are defined over the same event set T and hence it would be necessary to expect any low-level sequence in different models $G_{l,k}$, to be mapped to the same high-level sequence. In other words, the crucial *feasibility* condition for the information mapping is that for all $1 \leq i,j \leq n$ we have:

$$\omega_i(s) = \omega_j(s) \ \text{ for } \ s \in L(G_{l,i}) \cap L(G_{l,j}). \tag{11}$$

The feasibility condition in (11) requires any string s which is common among several models, to be transmitted the same in all those models. If the feasibility condition in (11) is satisfied for all $1 \leq i,j \leq n$ it would be possible to define an information map $\theta : \bigcup_k L(G_{l,k}) \longmapsto T^*$ as follows:

$$\theta(s) = \theta_k(s) \quad \text{for any k such that } s \in L(G_{l,k}) \tag{12}$$

Obtaining a feasible information map θ for a set of Moore automata $G_{l,k}$ is guaranteed if we can ensure the feasibility condition of (11) holds among the strings s common between the models $G_{l,k}$. To verify (11) for any two Moore automata $G_{l,i}$ and $G_{l,j}$ for $1 \leq i,j \leq n$, let $\hat{G}_{i,j} = G_{l,i} \times G_{l,j}$ be the meet product of them. Then condition (11) holds if and only if in every state (q_i,q_j) of $\hat{G}_{i,j}$ we have $\omega_i(q_i) = \omega_j(q_j)$ where ω_i is the output map of the model $G_{l,i}$.

Remark 3.1: In the continuation of this work we assume that the Moore automata $G_{l,k}$, with the individual output maps ω_k, satisfy the feasibility condition in (11) and hence we only refer to an information map θ defined in (12) for the purpose of exchanging the information between the two layers. Therefore, we can rewrite (10) as:

$$L(G_{h,k}) = \theta(L(G_{l,k})). \tag{13}$$

■

B. Joint-OCC Property

The HRSC problem, by definition, is based on RSC and HSC problems. In HSC problem, a Moore automaton $G_{l,k}$ should satisfy the OCC property and in RSC problem the models should agree on the controllability of the events. Note that this agreement at the low-level is obtained by the definition of the events for each model while at the high-level it is obtained by guaranteeing that the feasibility property (11) holds in the system and hence a unique information map θ exists for it. Nevertheless, we are not yet guaranteed that any automaton G_{hi} whose closed behavior is $L(G_{hi}) = \bigcup_k L(G_{h,k})$ is well-defined. In other words, we should make sure that such G_{hi} has a correspondent model at the low-level, say G_{lo} where $L(G_{lo}) = \bigcup_k L(G_{l,k})$, and we have $L(G_{hi}) = \theta(L(G_{lo}))$. This would happen if we could show OCC property holds for the $L_{lo} = \bigcup_k L(G_{l,k})$.

Definition 3.2: Consider a set of Moore automata $G_{l,k} = (Q_k, \Sigma_k, T_o, \delta_k, \omega_k, q_{0,k}, Q_k)$ $(k=1,..,n)$. Let $L_{lo} = \bigcup_k L(G_{l,k})$ and L_{voc} be the strings in L_{lo} with vocal ends. We say $G_{l,k}$ $(k=1,..,n)$ satisfy the *joint-OCC* property if for all $s \in L_{voc}$ we have (6) in definition 2.3. ■

It follows from the feasibility condition that the joint-OCC property is well-defined. It is shown in the following how L_{lo} can be presented with a Moore automaton.

Remark 3.3: Consider a set of Moore automata $G_{l,k}$ $(k=1,..,n)$ with the output maps $\omega_k : L(G_{l,k}) \longmapsto T_o$. Assuming that all the states in $G_{l,k}$'s are marked, we construct G_{lo} as follows:

$$G_{lo} = trim[(G^{co}_{l,1} \times ... \times G^{co}_{l,n})^{co}] \tag{14}$$

where $trim(.)$ denote the trim function and G^{co} stands for the complement of automaton G w.r.t the set $\Sigma = \bigcup_k \Sigma_k$ respectively. Each state of G_{lo} would be of the form $q = (q_1,...,q_n)$ in which at least of the elements q_i's belongs to the state set Q_i of $G_{l,i}$. That means any sequence leading to a state q in G_{lo}, leads to q_i in $G_{l,i}$. Define the output map $\omega(s) = \omega_i(s)$. It follows from the feasibility condition in (11) that this definition is well-defined. The closed behavior of G_{lo} with above structure is $L(G_{lo}) = L_{lo}$ and besides, contains the information outputs of $G_{l,k}$'s. ■

Lemma 3.4: Consider the Moore automata $G_{l,k}$ $(k=1,..,n)$ and let G_{lo} be defined in remark 3.3. Then $G_{l,k}$'s satisfy the *joint-OCC* property if and only if G_{lo} satisfies the OCC property. ■

Proposition 3.5: Consider the Moore automata $G_{l,k}$ $(k=1,..,n)$. Then $G_{l,k}$'s satisfy the *joint-OCC* property if and only if each $G_{l,k}$ satisfies the OCC property. ■

Corollary 3.6: $G_{l,k}$ $(k=1,..,n)$ are individually OCC if and only if $G_{lo} = \bigcup_k G_{l,k}$, is OCC. ■

IV. Supervision Implementation at the Low-Level

Let $G_{h,k}$ be the models at the high-level whose closed language is $L(G_{h,k}) = \theta(L(G_{l,k}))$. We assume the desired behavior is given by $E'_{h,k}$, S_{hi} solves the robust supervisory control problem and $E_{h,k} = L(S_{hi}/G_{h,k})$ is the system under supervision at the high-level. Also let $E_{hi} = \bigcup_k E_{h,k}$, $E_{lo} = \theta^{-1}(E_{hi})$ and finally $E'_{l,k} = \theta^{-1}(E'_{h,k})$ be the corresponding legal language for each $L(G_{l,k})$ at the low-level.

Let G_{hi} be any automaton at the high-level whose closed behavior is $L(G_{hi}) = \bigcup_k L(G_{h,k})$. Theorem 2.1 implies $L(S_{hi}/G_{hi}) = E_{hi}$ and E_{hi} is controllable w.r.t $L(G_{hi})$. Also let G_{lo} be a Moore automata which is given by (14) in Remark 3.3 for the set of models $G_{l,k}$. By definition of G_{hi} and G_{lo} we have $L(G_{hi}) = \theta(L(G_{lo}))$. Now from Theorem 2.4, we conclude that if the high-level supervisor S_{hi} is implemented by a disabled-event map Δ_{hi} then there exists a low-level supervisor S_{lo} which is implemented by a disabled-event map Δ_{lo} given in (7). We show S_{lo} solves the robust supervisory control at the low level for the set of models $G_{l,k}$ and the specifications $E'_{l,k}$.

Theorem 4.1: Consider a set of models $G_{l,k}$ which are jointly OCC. Then with the above notations, we have $L(S_{lo}/G_{l,k}) = E^{\uparrow}_{lo} \cap L(G_{l,k}) \subseteq E'_{l,k}$. ■

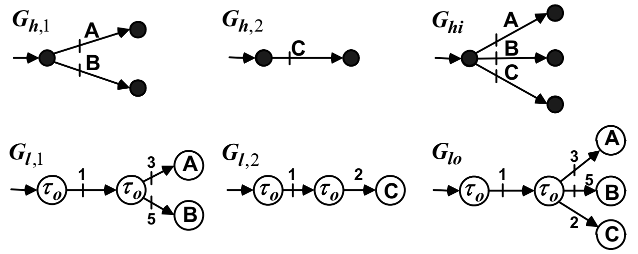


Fig. 3. $G_{l,1}$ and $G_{l,2}$ satisfy the SOCC condition individually but the model G_{lo}, which describes the union of them, does not.

Example 4.2: Fig. 3 shows a system in which each model $G_{l,k}$ satisfies the OCC condition. Here the odd numbers refer to controllable events while the even numbers refer to uncontrollable events. Let the high-level specifications $E'_{h,1} = \overline{A+B}$ and $E'_{h,2} = \{\varepsilon\}$ be given. The system under supervision at the high-level would be $L(S_{hi}/G_{hi}) = E_{hi} = \overline{A+B}$. The image of E_{hi} at the low-level would be $E_{lo} = \{\varepsilon, "1", "1.3", "1.5"\}$ and the controllable image of E_{hi} is $L(S_{lo}/G_{lo}) = E_{lo}^{\uparrow} = \{\varepsilon\}$. Thus, the systems under supervision at the low-level would be $L(S_{lo}/G_{l,i}) = \{\varepsilon\} \cap L(G_{l,i}) = \{\varepsilon\}$ for $i = 1, 2$. ■

We would like the low-level implementation of the supervisory control $(L(S_{lo}/G_{l,k}))$ to map to the high-level supervisor's expectations $(L(S_{hi}/G_{h,k}))$ and refer to this property as *Robust Hierarchical Consistency*.

Definition 4.3: We say the *robust hierarchical consistency* property holds in the system if $\theta(L(S_{lo}/G_{l,k})) = L(S_{hi}/G_{h,k})$ for $(k = 1, .., n)$. ■

In the robust control problem, we have to make sure that implementation of a high-level command to disable an event does not produce unintended consequences in any possible plant. To ensure this, we bring in a Joint SOCC property.

Definition 4.4: The Moore generators $G_{l,k}$ $(k = 1, .., n)$ are Jointly SOCC if i) they are jointly OCC and ii) in the reachability tree of $\bigcup_k L(G_{l,k})$, no two red vocal nodes are partners. ■

In the following proposition, we show if $G_{l,k}$ $(k = 1, .., n)$ are jointly SOCC then $G_{l,k}$ are individually SOCC. The reverse is not true as shown in an example following Proposition 4.5.

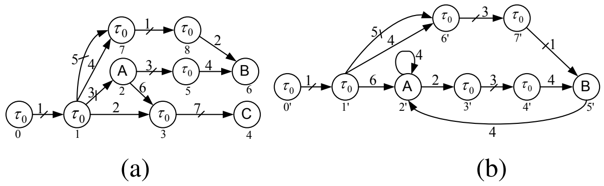
Proposition 4.5: Let a set of Moore generators $G_{l,k}$ be given for $(k = 1, .., n)$. If $G_{l,k}$ $(k = 1, .., n)$ are jointly SOCC then $G_{l,k}$ $(k = 1, .., n)$ are individually SOCC.

Later in Example 4.9, we show if $G_{l,k}$'s are not jointly SOCC, how they can be modified to become jointly SOCC.

Example 4.6: With the given models in example 4.2, $G_{l,1}$ and $G_{l,2}$ satisfy the SOCC condition but in G_{lo}, two states (A, C) (or (B, C)) are partners. ■

In the following we show how the *robust hierarchical consistency* property holds in a HRSC system.

Theorem 4.7: Let the low-level Moore automata $G_{l,k}$ $(k = 1, .., n)$ be given and also $G_{h,k}$, $E'_{h,k}$, S_{hi}, $E_{h,k}$, E_{hi}, E_{lo}, $E'_{l,k}$, G_{lo} and S_{lo} be as previously defined in Theorem 4.1. If $G_{l,k}$ $(k = 1, .., n)$ are jointly SOCC then we have $\theta[L(S_{lo}/G_{l,k})] = E_{h,k}$. ■



(a)

(b)

Fig. 4. (a): G_1 and (b): G_2 with vocal states

Example 4.8: The system in Example 4.6 does not satisfy the robust hierarchical consistency. Specifically we have $\theta(L(S_{lo}/G_{l,1})) = \{\varepsilon\} \neq E_{h,1}$. It follows from Example 4.6 that $G_{l,i}$ for $i = 1, 2$ do not satisfy the joint SOCC property and hence with the given specifications $E_{h,1}$ and $E_{h,2}$ at the high-level, violating the *robust hierarchical consistency* is justified. Again note that in general the joint SOCC property is a sufficient, and not necessary, condition for robust hierarchical consistency. ■

Example 4.9: The theory developed in this paper is illustrated through this example. Figs. 4.(a) and (b) show two low-level Moore automata G_1 and G_2 with individually assigned output maps ω_i for $i = 1, 2$. It can be checked that the feasibility condition holds for them. Neither G_1 nor G_2 is OCC and only G_1 is SOCC. Fig. 5 shows the Moore model $G = G_1 \cup G_2$ whose equivalent output map ω (see remark 3.3) has been refined so that G is SOCC and OCC. The following changes have occurred in G: the state reached by $s = 1$ has been assigned a new output D $(\omega(1) = D)$ to make the SOCC property hold in G; the state reached by the sequence $s' = 1.6$ has been renamed to A_{uc} $(\omega("1.6") = A_{uc})$ to reflect the uncontrollable behavior of the last silent segment of the sequence $s' = 1.6$; and finally sequences $s_1 = 1.4$ and $s_2 = 1.5$, whereas in fig. 4 in which we have $\delta(q_0, "1.4") = \delta(q_0, "1.5")$, reach different states (see OCC algorithm in [13]). Since G is SOCC, we can show that $\hat{G}_1 = G \times G_1$ and $\hat{G}_2 = G \times G_2$ will be jointly SOCC. (Note the state of $\hat{G}_i$ will be of the form (x, y) where x and y are the states of G and G_i, respectively and the output at (x,y) is taken to be equal to the output of G at x). The results are shown in Fig. 6. The abstracted models $G_{h,1}$ and $G_{h,2}$ are obtained from the models in fig. 6 and are shown in Fig. 7. Suppose $E'_{h,1} = L(G_{h,1}) - \{DC\}$ and $E'_{h_2} = \bigcup_n \overline{\{DA^n_{uc}B\}}$ for $n \geq 1$. For the given set of high-level models and legal languages there would be a robust supervisor S_{hi} for which the systems under supervision could be shown by Fig. 8.(a) and (b). Theorem 4.1 guarantees that $E^{\uparrow}_{l,i} \cap L(G_i)$ for $i = 1, 2$ is the language of G_i under the supervision at the low-level where $E_{l,i} = \theta^{-1}(E_{h,i})$ as previously elaborated. Figs. 9.(a) and (b) illustrate these facts. Note that in the states (3) and (10) of the graph of (S_{lo}/G_1) in Fig. 9.(a), the plant G1 has reached its state (3) through strings "1.3.6" and "1.2", respectively. In the first case, we allow "C" to occur and in the other case, we disable C (see (7)). Since the the models in Fig. 6 are jointly SOCC, Theorem 4.7 guarantees that the robust hierarchical consistency defined in Definition 4.3 holds in the system. It is easy to check that

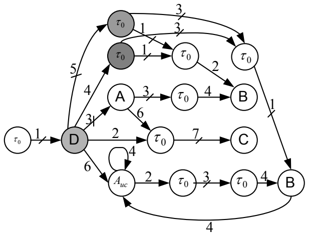
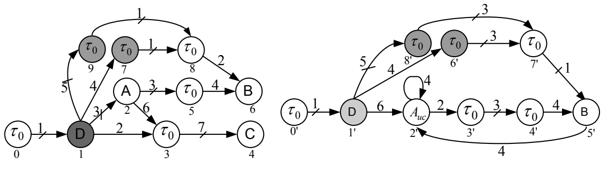


Fig. 5. union of G_1 and G_2, $G = G_1 \cup G_2$, has been enforced to satisfy the SOCC and OCC conditions.



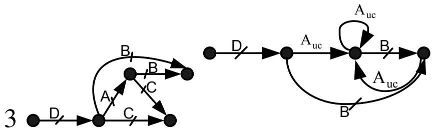
(a) (b)

Fig. 6. (a): $\hat{G}_1$ and (b) $\hat{G}_2$ satisfy the joint-OCC and joint-SOCC conditions

the abstractions of the low-level languages under supervision given by the automata in Fig. 9, are given by the languages of the automata in Fig. 8, i.e. $\theta[L(S_{lo}/\hat{G}_i)] = E_{h,i}$ for $i = 1,2$. ■

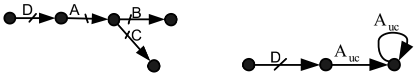
V. Conclusion

In this paper a hierarchical solution for the problem of robust supervisory control of a finite family of DES plants was derived based on Zhong and Wonham approach. Joint-OCC and joint-SOCC properties as the extensions of the OCC and SOCC properties were developed as sufficient conditions to guarantee the robust hierarchical consistency. We showed the robust supervisor which is designed for the high-level yields the maximum controllable behavior at the low-level while at the same time, robust hierarchical consistency holds in the system.



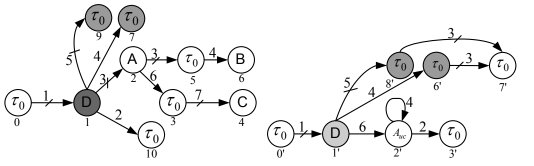
(a) (b)

Fig. 7. (a): high-level model $G_{h,1}$ (b):high-level model $G_{h,2}$



(a) (b)

Fig. 8. (a): $G_{h,1}$ under supervision, $E_{h,1} = L(S_{hi}/G_{h,1})$ (b): $G_{h,1}$ under supervision, $E_{h,2} = L(S_{hi}/G_{h,2})$



(a) (b)

Fig. 9. (a): G_1 under supervision, $L(S_{lo}/G_1)$ (b): G_2 under supervision, $L(S_{lo}/G_2)$

References

[1] S. Bourdon, M. Lawford, and W. Wonham. Robust nonblocking supervisory control of discrete-event systems. *IEEE Transactions on Automatic Control*, 50(12):2015–2020, 2005.

[2] Y. Brave and M. Heymann. Control of discrete-event systems modeled as hierarchical state machines. *IEEE Transactions on Automatic Control*, 38(12):1803–1819, 1993.

[3] P.E. Caines, V. Gupta, and G. Shen. The hierarchical control of st-finite state machines. *In Proceedings of the 36th Conference on Decision and Control*, pages 3584–3589, San Diego, California, USA, December 1997.

[4] J.E.R Cury and B.H. Krogh. Robustness of supervisors for discrete-event systems. *IEEE Transactions on Automatic Control*, 44(2):376–379, 1999.

[5] K. C.Wong and W. M.Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems:Theory and Applications*, 6(3):241–273, 1996.

[6] S.G. Kim, K.H. Cho, and J.T. Lim. Hierarchical supervisory control of discrete event systems based on h-observability. *IEE Proceedings, Control Theory and Applications*, 150(2):179–182, March 2003.

[7] R.J. Leduc, M. Lawford, and W.M. Wonham. Hierarchical interface-based supervisory control-part ii: parallel case. *IEEE Trans. Automatic Control*, 50(9):1336–1348, Sept. 2005.

[8] F. Lin. Robust and adaptive supervisory control of discrete event systems. *IEEE Transactions on Automatic Control*, 38(12):1848–1852, 1993.

[9] S.J. Park and J.T. Lim. Hierarchical supervisory control of discrete event systems with model uncertainty. *International Journal of Systems Science*, 32(6):739–744, 2001.

[10] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event systems. *SIAM Journal of Control Optimization,*, 25(1):206–230, 1987.

[11] A. Saboori and S. Hashtrudi Zad. Fault recovery in discrete event systems. *Proc. Computational Intelligence: Methods and Applications, CIMA'05 (ICSC/IEEE)*, Istanbul, Turkey, Dec 2005.

[12] A. Saboori and S. Hashtrudi Zad. Robust nonblocking supervisory control of discrete-event systems under partial observation. *Systems and Control Letters*, 55(10):839–848, Oct 2006.

[13] W. M. Wonham. Supervisory control of discrete-event systems, lecture notes. *Department of Electrical and Computer Engineering*, University of Toronto, 2005.

[14] W.M. Wonham and P.J. Ramadge. Modular supervisory control of discrete-event systems. *Journal of Mathematics of Control, Signals, and Systems*, 1(1):13–30, 1988.

[15] H. Zhong and W.M. Wonham. On the consistency of hierarchical supervision in discrete-event systems. *IEEE Transactions on Automatic Control*, 35(10):1125–1134, 1990.