Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Top-Down Nested Supervisory Control of State-Tree Structures Based on State Aggregations [⋆]

Xi Wang [*,**] **Thomas Moor** [**] **Zhiwu Li** [*,***]

[*] *School of Electro-Mechanical Engineering, Xidian University, China (e-mail: wangxi@xidian.edu.cn, zhwli@xidian.edu.cn).*
[**] *Lehrstuhl für Regelungstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany (e-mail: lrt@fau.de).*
[***] *Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau.*

Abstract: With a structured state space, state-tree structures (STS) are a powerful framework to model hierarchical finite state machines (HFSM). The boundary consistency property of STS endows them a compact and neatly representation. In this study, by naturally decomposing an STS into a set of STS nests in a top-down nested approach and finding a supervisor for each, the boundary consistency property is extended to the supervisory control of STS. As a consequence, the state spaces for both the system model and optimal supervisor are significantly reduced. Two examples are provided, in which the state space of a large scale HFSM example is reduced from 10^{24} to 2×10^{18}.

Keywords: Discrete-event system, nonblocking supervisory control, state-tree structure, symbolic computation, nested state feedback control.

1. INTRODUCTION

Hierarchical finite state machines (HFSM) (Marchand & Gaudin (2002); Gaudin & Marchand (2004, 2005)) were first developed in Harel (1985, 1987) as finite state machines (FSM) (Wonham & Cai (2013); Ramadge & Wonham (1985)) with multi-levels. As stated in Alur et al. (1999), a superstate in an HFSM can also be other machines. Traditionally, the calculation of the supervisor for an HFSM is obtained by two steps: flatting an HFSM to be an equivalent FSM, and which is followed by using the standard supervisory control of discrete-event systems (DES) (Wonham & Cai (2013); Ramadge & Wonham (1985)). For the purpose of managing the notorious state explosion problem caused by flatting HFSM, state-tree structures (STS) are proposed in Ma & Wonham (2005, 2006) to build HFSM in a compact and natural model. A main feature of STS is that it satisfies *boundary consistency*: without changing the input/output transitions of a given level, several lower level structures can be "plugged" into its superstates. *Binary decision diagram* (BDD) (Bryant (1986)) is utilized in the supervisory control of STS as a powerful computational representation of predicates, based on which the state explosion problem faced by the supervisory control of DES is managed.

Several theoretical extensions and applications of STS have been made. The modular supervisory control of an STS is studied in Chao et al. (2013). Supervisor localization based on STS is proposed in Cai & Wonham (2015) to calculate the controller of a controllable event by considering the agent's neighborhood information only. The research in Jiao et al. (2017) studies the symmetry of STS with parallel components. The supervisory control of STS with partial observation is investigated in Gu et al. (2018) and Gu et al. (2019). In Wang et al. (2019), the nonblocking supervisory control of STS with conditional-preemption matrices is assigned.

In this study, the boundary consistency of STS is extended to the supervisory control of STS. Given an STS with a set of superstates assigned, it is naturally decomposed into a set of STS nests rooted by superstates, which describe the system behavior locally. As a consequence, instead of calculating the optimal behavior of an STS monolithically, a top-down nested approach is presented to calculate the optimal behavior of each STS nest. The main contributions of this study are: 1) based on the superstates, the STS nests describing the system behavior on each hierarchical level are formally defined; 2) the subordination relation of the STS nests is investigated; and 3) the state spaces for both the system model and optimal supervisor are significantly reduced. Two examples are provided in this study. For a large scale example AIP (Ma & Wonham (2005, 2006)), its state space is reduced from 10^{24} to 2×10^{18}.

The rest of this paper is organized as follows. Section 2 presents the STS terminology used throughout the paper.

[⋆] This work was supported in part by the Natural Science Foundation of China under Grant No. 61703322, the Alexander von Humboldt Foundation, the Science and Technology Development Fund, MSAR, under Grant No. 0012/2019/A1, and the Fundamental Research Funds for the Central Universities under Grant Nos. XJS200403 and JBF180401. (Corresponding author: Xi Wang.)

The nested structure of STS is studied in Section 3. The nested supervisory control of STS is presented in Section 4. Two case studies are presented in Section 5 to demonstrate the nested supervisory control of STS. Further relevant issues are discussed in Section 6. Finally, conclusions and future work are presented in Section 7.

2. STS PRELIMINARIES

Relevant preliminaries on the supervisory control of STS are summarized from Ma & Wonham (2005, 2006). An STS is a six-tuple $\mathbf{G} = (\mathbf{S}T, \mathcal{H}, \Sigma, \Delta, \mathbf{S}T_0, \mathcal{S}T_m)$, where $\mathbf{S}T$ is a *state-tree* organizing the the state space of an STS hierarchically; $\mathcal{H}$ is the set of *holons* (finite automata); Σ is the finite event set appeared in $\mathcal{H}$; Δ is the *global function* $\mathcal{S}T(\mathbf{S}T) \times \Sigma \to \mathcal{S}T(\mathbf{S}T)$, where $\mathcal{S}T(\mathbf{S}T)$ is the set of all *sub-state-trees*; $\mathbf{S}T_0$ is the *initial state-tree*; and $\mathcal{S}T_m$ is the set of *marker state-trees.* All the *basic state-trees* are denoted by $\mathcal{B}(\mathbf{S}T)$, in which each element $b \in \mathcal{B}(\mathbf{S}T)$ corresponds to a "flat" system state.

A state-tree consists of three types of states: AND, OR, and SIM, in which AND and OR denote the superstates and SIM represents the simple states. In a state-tree $\mathbf{S}T = (X, x_0, \mathcal{T}, \mathcal{E})$, X is the finite *structured state set*; $x_o \in X$ is the *root state*; $\mathcal{T} : X \to \{AND,\ OR,\ SIM\}$ is the *type function*; and $\mathcal{E} : X \to 2^X$ is the *expansion function.* The *reflexive* and *transitive closure* of $\mathcal{E}$ is written as $\mathcal{E}^*$. Then $\mathcal{E}^+(x) := \mathcal{E}^*(x) - \{x\}$ represents the set of all *descendants* of x. In an $\mathbf{S}T$, let $x, y \in X$. $x \leq y$ (resp., $x < y$) iff $y \in \mathcal{E}^*(x)$ (resp., $y \in \mathcal{E}^+(x)$). Let $x < y$. Define that y is *AND-adjacent* to x, i.e., $x <_\times y$ iff $x < y \ \&\ \mathcal{T}(x) = AND \wedge (\forall z) x < z < y \Rightarrow \mathcal{T}(z) = AND.$ Based on AND and OR superstates, the state space of an STS can be decomposed into several successive layers in a top-down format, which consist of *Cartesian products* and *disjoint unions*, respectively. In a well-formed state-tree, all the leaf states are simple states.

A holon is a five-tuple $H := (X, \Sigma, \delta, X_0, X_m)$, where X is the *nonempty state set* that can be partitioned into a (possibly empty) *external state set* X_E and an *internal state set* X_I, i.e., $X = X_E \dot{\cup} X_I$ with $X_E \cap X_I = \emptyset$; Σ is the *event set* that can be partitioned into a *boundary event* set Σ_B and an *internal event set* Σ_I, i.e., $\Sigma = \Sigma_B \dot{\cup} \Sigma_I$. Σ can also be partitioned into the sets of *controllable* and *uncontrollable* events, i.e., $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$; The *transition structure* $\delta : X \times \Sigma \to X$ is a *partial function.* We write $\delta(x, \sigma)!$ if $\delta(x, \sigma)$ is defined. $X_0 \subseteq X_I$ is the *initial state set*; and $X_m \subseteq X_I$ is the *terminal state set.* The interval behavior of a holon is assigned to an *OR superstate*, which describes the local behavior of an STS $\mathbf{G}$ Ma & Wonham (2006). In a holon, X_0 (resp., X_m) contains the *target* (resp., *source*) *states* of the *boundary transitions* iff $X_E \neq \emptyset$, i.e., a higher level holon exists.

Intuitively, a predicate P is defined based on $\mathcal{B}(\mathbf{S}T)$, such that P: $\mathcal{B}(\mathbf{S}T) \to \{0, 1\}$. The truth-value 1 (resp., 0) represents logical *true* (resp., *false*). Formally, $P(b) = 1$ is represented by $b \models P$. Propositional logic operators are defined by: 1) $(\neg P)(b) = 1$ iff $P(b) = 0$; 2) $(P_1 \wedge P_2)(b) = 1$ iff $P_1(b) = 1$ and $P_2(b) = 1$; and 3) $(P_1 \vee P_2)(b) = 1$ iff $P_1(b) = 1$ or $P_2(b) = 1$. A predicate P is identified by a set of basic-state-trees if $\mathcal{B}_P := \{b \in \mathcal{B}(\mathbf{S}T) | P(b) = 1\} \subseteq \mathcal{B}(\mathbf{S}T)$. Let P_0 and P_m denote the predicate identified by the initial state-tree $\mathbf{S}T_0$ and the marker state-tree set $\mathbf{S}T_m$, respectively. Then we have $\mathcal{B}_{P_0} := \{b \in \mathcal{B}(\mathbf{S}T) | b \models P_0\}$ and $B_{P_m} := \{b \in \mathcal{B}(\mathbf{S}T) | b \models P_m\}$. The set of all predicates on $\mathcal{B}(\mathbf{S}T)$ is defined by $Pred(\mathbf{S}T)$. The top and bottom elements of a predicate are *true* ($\top$) and *false* ($\perp$), respectively. For an STS $\mathbf{G}$ and a given predicate P, via all the basic-state-trees (T) satisfying P, the reachability (sub)predicate $R(\mathbf{G}, P)$ holds on a sequence of T can be reached from some $b_0 \models P \wedge P_0$. Dually, $CR(\mathbf{G}, P)$, namely the coreachability predicate, holds all the T that can reach some $b_m \models P \wedge P_m$ by a sequence of T satisfying P.

3. NESTED STRUCTURE OF STS

An STS is a framework to model hierarchical DES in a compact form. Given an STS with a set of superstates assigned, we can naturally decompose it into a set of STS nests rooted by superstates, which describe the system behavior locally.

3.1 STS Nests

The state aggregation of each superstate is define below.

Definition 1. [State Aggregation] Let $\mathbf{S}T = (X, x_0, \mathcal{T}, \mathcal{E})$ be a state-tree. The *state aggregation* $X_\mathcal{A} : X \to 2^X$ in a state-tree is defined by

$$X_\mathcal{A}(x) := \begin{cases} \{z | z \in \mathcal{E}(x)\}, & \text{if } \mathcal{T}(x) = OR \\ \{z | z \in X_\mathcal{A}(y), & \\ \quad \mathcal{T}(y) = or, x <_\times y\}, & \text{if } \mathcal{T}(x) = AND \end{cases}.$$

In the case that $X_\mathcal{A}(y) \subset X_\mathcal{A}(x)$ with $x < y$, the calculation of state-aggregation $X_\mathcal{A}(y)$ is discarded. The remains partition the structured state set of an STS. $\diamond$

Definition 2. [STS Nest] An STS nest $\underline{\mathbf{G}}^x$ rooted by a superstate x describes the local behavior in x, which is represented by a six-tuple $\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_\mathcal{A}(x), \Sigma_\mathcal{A}(x), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$, where

- $\underline{\mathbf{S}T}^x$ is a state-tree with a root x and terminated at the state aggregation $X_\mathcal{A}(x)$. In an STS $\mathbf{G}$, given a state-tree $\mathbf{S}T^x$ rooted by x, $\underline{\mathbf{S}T}^x$ is obtained by removing all the descendents of y with respect to $X_\mathcal{A}(x) < X_\mathcal{A}(y)$;
- $\mathcal{H}_\mathcal{A}(x)$ is the *holon aggregation* of superstate x. Formally, $\mathcal{H}_\mathcal{A}(x) := \{H^y | X_I^y \subseteq X_\mathcal{A}(x)\}$;
- $\Sigma_\mathcal{A}(x)$ is the *event aggregation* of superstate x. Formally, $\Sigma_\mathcal{A}(x) := \{\sigma | \sigma \in \Sigma_I^y, H^y \in \mathcal{H}_\mathcal{A}(x)\}$;
- $\underline{\Delta}^x$ is the *nested transition structure* of $\underline{\mathbf{G}}^x$, which will be defined later;
- $\underline{\mathbf{S}T_0}^x$ is the *initial-state-tree* of $\underline{\mathbf{G}}^x$ with respect to $\mathcal{V}(\underline{\mathbf{S}T_0}^x) = \{z | z \in X_0^y, H^y \in \mathcal{H}_\mathcal{A}(x)\}$; and
- $\underline{\mathbf{S}T_m}^x$ is the *marker-state-tree set* of $\underline{\mathbf{G}}^x$ with respect to $\mathcal{V}(\underline{\mathbf{S}T_m}^x) = \{z | z \in X_m^y, H^y \in \mathcal{H}_\mathcal{A}(x)\}$. $\diamond$

By starting from the root state of an STS $\mathbf{G}$, it is naturally decomposed into a set of STS nests by Algorithm 1.

Definition 3. [Subordination] STS nest $\underline{\mathbf{G}}^y$ is subordinate to $\underline{\mathbf{G}}^x$ if $y \in X_\mathcal{A}(x)$. Formally, $\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y$. $\diamond$

Example.

We take the transfer line (Wonham & Cai (2013); Ma & Wonham (2005, 2006)) shown in Fig. 1 as an example.

Algorithm 1 STS nest calculation

```
Input: An STS $\mathbf{G}$ with root state $x_0$.
Output: A set $\mathcal{D}$ of STS nests $\underline{\mathbf{G}}$.
1. Calculate $X_{\mathcal{A}}(x_0)$ and $\underline{\mathbf{G}}^{x_0}$;
2. Put state $y \in X_{\mathcal{A}}(x_0)$ w.r.t. $\mathcal{T}(y) \neq SIM$ in a set $\mathcal{C}$;
3. while $\mathcal{C} \neq \emptyset$ do
4.     Choose an element $y$ from $\mathcal{C}$;
5.     $\mathcal{C} := \mathcal{C} \setminus \{y\}$;
6.     Calculate $X_{\mathcal{A}}(y)$ and $\underline{\mathbf{G}}^{y}$;
7.     Put state $z \in X_{\mathcal{A}}(y)$ w.r.t. $\mathcal{T}(z) \neq SIM$ in $\mathcal{C}$;
8.     Put $\underline{\mathbf{G}}^{y}$ into $\mathcal{D}$;
9. end
10. return $\mathcal{D}$;
```

Suppose that the capacities of the buffers $B1$ and $B2$ are both one. The system behavior of machine $M1$ (resp., $M2$) is described by two holons, in which the operations in superstate $M1_1$ (resp., $M2_1$) are depicted in the low level holons. The corresponding state-tree and holons are shown in Figs. 2 and 3, respectively. The events denoted by odd (resp., even) numbers are controllable (resp., uncontrollable).

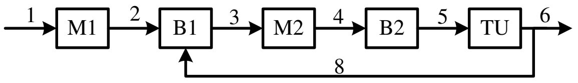

Fig. 1. Transfer line.

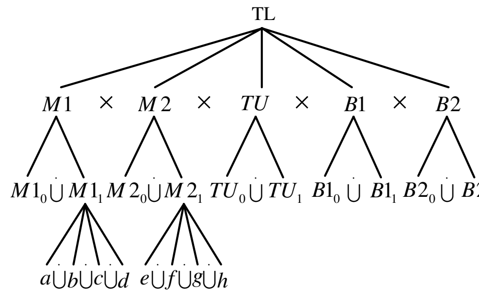
TL
M1 × M2 × TU × B1 × B2
M1_0 ∪̇ M1_1 M2_0 ∪̇ M2_1 TU_0 ∪̇ TU_1 B1_0 ∪̇ B1_1 B2_0 ∪̇ B2_1
a ∪̇ b ∪̇ c ∪̇ d e ∪̇ f ∪̇ g ∪̇ h

Fig. 2. State-tree of a transfer line after plug in.

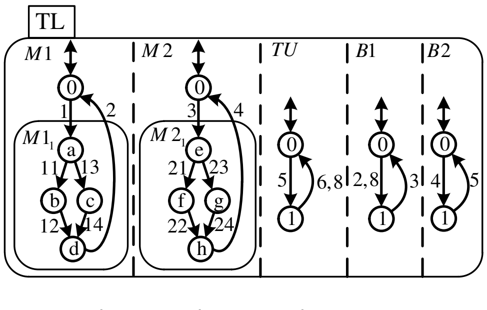

Fig. 3. Holons of a transfer line after plug in.

For the STS shown in Figs. 2 and 3, we have $\underline{\mathbf{G}}^{TL} < \underline{\mathbf{G}}^{M1_1}$ and $\underline{\mathbf{G}}^{TL} < \underline{\mathbf{G}}^{M2_1}$. The holons are divided into three holon families shown in Figs. 4 and 5. The system behaviors in $\underline{\mathbf{G}}^{M1_1}$ and $\underline{\mathbf{G}}^{M2_1}$ are not considered while synthesizing the supervisor for the top level $\underline{\mathbf{G}}^{TL}$. □

3.2 Nested Transition Structure in STS

Generally, given an STS $\mathbf{G}$, suppose that $\underline{\mathbf{G}}^{y}$ is subordinate to $\underline{\mathbf{G}}^{x}$, i.e., $\underline{\mathbf{G}}^{x} < \underline{\mathbf{G}}^{y}$. We have holon aggregations $\mathcal{H}_{\mathcal{A}}(x)$ and $\mathcal{H}_{\mathcal{A}}(y)$ as two holon families. In $\mathcal{H}_{\mathcal{A}}(x)$, superstate y is replaced by a simple state with the same name. In order to integrate the system behavior of $\underline{\mathbf{G}}^{y}$ into $\underline{\mathbf{G}}^{x}$, we require that

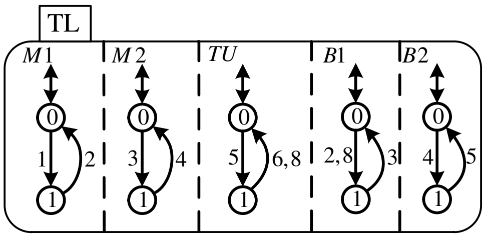

Fig. 4. Holon family of $\underline{\mathbf{G}}^{TL}$.

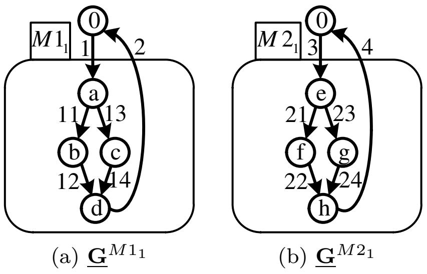
M1_1
M2_1
(a) $\underline{\mathbf{G}}^{M1_1}$
(b) $\underline{\mathbf{G}}^{M2_1}$

Fig. 5. Holon families of $\underline{\mathbf{G}}^{M1_1}$ and $\underline{\mathbf{G}}^{M2_1}$.

- while the system arrives state y in $\underline{\mathbf{G}}^{x}$, the system enters $\underline{\mathbf{ST}_0}^{y}$ automatically;
- after entering $\underline{\mathbf{G}}^{y}$, the process in $\underline{\mathbf{G}}^{x}$ is paused temporarily; and
- while the system is ready to leave $\underline{\mathbf{G}}^{y}$, the events defined at state y in $\underline{\mathbf{G}}^{x}$ are eligible to occur.

Let σ in Σ be an event in an STS $\mathbf{G}$. In Ma & Wonham (2005, 2006), the *largest eligible state-tree* and *largest next state-tree* of σ, denoted by $Elig_{\mathbf{G}}(\sigma)$ and $Next_{\mathbf{G}}(\sigma)$, are proposed to build its *forward* and *backward transitions*, respectively. Let $\sigma \in \Sigma_{\mathcal{A}}(x)$. Similarly, the *largest nested eligible state-tree* and *largest nested next state-tree* in $\underline{\mathbf{G}}^{x}$, denoted by $Elig_{\underline{\mathbf{G}}^{x}}(\sigma)$ and $Next_{\underline{\mathbf{G}}^{x}}(\sigma)$, respectively, are obtained. As a consequence, the *nested forward/backward transition structures* denoted by $\underline{\Delta}^{x}/\underline{\Gamma}^{x}$ are built.

Similarly, at a state-tree $T \in \mathcal{ST}(\underline{\mathbf{ST}}^{x})$, the forward (resp., backward) *transition relation* $\underline{\Delta}^{x}$ (resp., $\underline{\Gamma}^{x}$) corresponding to event σ is defined based on replace_source$_{\underline{\mathbf{G}}^{x},\sigma}$ (resp., replace_target$_{\underline{\mathbf{G}}^{x},\sigma}$) operations on $T \wedge Elig_{\underline{\mathbf{G}}^{x}}(\sigma)$ (resp., $T \wedge Next_{\underline{\mathbf{G}}^{x}}(\sigma)$). According to Ma & Wonham (2005, 2006), function replace_source$_{\underline{\mathbf{G}}^{x},\sigma}$ (resp., replace_target$_{\underline{\mathbf{G}}^{x},\sigma}$) replaces the source (resp., target) states of event σ appeared in T by the corresponding target states simultaneously.

Example.

For the STS shown in Figs. 4 and 5, while the system arrives state $M1_1$ in $\underline{\mathbf{G}}^{TL}$, the system enters $\underline{\mathbf{ST}_0}^{M1_1}$ automatically and the behavior in $\underline{\mathbf{G}}^{TL}$ is paused. After the system arrives $\underline{\mathbf{ST}_m}^{M1_1}$ in $\underline{\mathbf{G}}^{M1_1}$, event 2 in $\underline{\mathbf{G}}^{TL}$ is eligible to occur. In Fig. 4, at the active state set [1] $\{M2_0, B1_1\}$, we have $\underline{\Delta}^{TL}(\{M2_0, B1_1\}, 3) = \{M2_1, B1_0\}$. □

3.3 Predicate Representation

The transition relation structures and the state space of all the STS nests are encoded in predicates by function $\Theta : \mathcal{ST}(\underline{\mathbf{ST}}^x) \to Pred(\underline{\mathbf{ST}}^x)$ defined in Ma & Wonham (2005, 2006). [2] Let $\underline{\mathbf{G}}^x = (\underline{\mathbf{ST}}^x, \mathcal{H}_{\mathcal{A}}(x), \Sigma_{\mathcal{A}}(x), \underline{\Delta}^x, \underline{\mathbf{ST}_0}^x, \underline{\mathbf{ST}_m}^x)$ be an STS nest. A predicate P^x defined on $\underline{\mathcal{B}(\mathbf{ST}^x)}$ is a function $P := \mathcal{B}(\underline{\mathbf{ST}}^x) \to (0,1)$. As a consequence, $\underline{\mathbf{G}}^x$ can be rewritten as $\underline{\mathbf{G}}^x = (\underline{\mathbf{ST}}^x, \mathcal{H}_{\mathcal{A}}(x), \Sigma_{\mathcal{A}}(x), \underline{\Delta}^x, \underline{P_0}^x, \underline{P_m}^x)$, in which $\underline{P_0}^x$ and $\underline{P_m}^x$ are the *initial predicate* and *marker predicate*, respectively.

4. NESTED SUPERVISORY CONTROL OF STS

In Ma & Wonham (2005, 2006), given a predicate P with respect to an STS $\mathbf{G}$, by the state feedback control (SFBC), the supremal element of weakly controllable and coreachable behavior of $\mathbf{G}$ (i.e., optimal behavior) $C = \sup\mathcal{C}^2\mathcal{P}(P)$ of $\mathbf{G}$ w.r.t. P is calculated. Instead of calculating the optimal supervisor of an STS monolithically, a top-down nested approach is presented in this section to calculate the optimal behavior of each STS nest.

4.1 Nested Supervisory Control

Suppose that $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ are two STS nests in an STS $\mathbf{G}$ w.r.t. $\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y$. By predefining specifications for each STS nest, we obtain predicates P^x and P^y, respectively. A top-down supervisor synthesis procedure is given in Algorithm 2 to calculate the optimal behavior of all the STS nests. The calculation of the optimal behavior w.r.t. P^y depends on the result for P^x. Finally, the global optimal behavior of $\mathbf{G}$ is obtained.

In Algorithm 2, Lines 1–3 calculate the supervisor for the STS nest $\underline{\mathbf{G}}^x$ on the top level. The control function of each controllable event on the top level is calculated in Line 3. Lines 4–21 define a recursive function Supcon$(\cdot)$ that calculate the supervisors of other STS nests. Suppose that $\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y$. Let holon $H^a \in \mathcal{H}_{\mathcal{A}}(x)$ and $H^b \in \mathcal{H}_{\mathcal{A}}(y)$ with $\sigma \in \Sigma_I^x \cap \Sigma_B^y \cap \Sigma_c$. In case that σ is disabled in $\underline{\mathbf{G}}^x$, then the basic-state-trees in $\underline{\mathbf{G}}^y$ containing the corresponding terminal states in H^b are defined as an illegal state in $\underline{\mathbf{G}}^y$, which is guaranteed by Lines 6–8. For each controllable event σ in $\Sigma_{\mathcal{A}}(y)$, Lines 9–11 calculate the control functions that contain all the (redundant) illegal behaviors of $\underline{\mathbf{G}}^x$ to provide the supremal permissive behavior for event σ. Line 13 checks the supremal behavior C^y for $\underline{\mathbf{G}}^y$. In case that C^y is empty or some basic-state-trees in $\underline{\mathbf{G}}^y$ containing the terminal states in $\underline{\mathbf{G}}^y$ are not reachable, which shows that the system model in $\underline{\mathbf{G}}^y$ is problemtic and the users should remodel it. Line 17 invokes function Supcon$(\cdot)$ recursively.

Algorithm 2 Nested Supervisory Control of STS

```
Input: A set of Nested STS $\underline{\mathbf{G}}$ with predefined predicates.
Output: Control functions for controllable events.
1. Compute $C^x = \sup\mathcal{C}^2\mathcal{P}(P^x)$ with $x = x_0$;
2. $N_{good} := \Theta(Next_{\mathbf{G}}(\sigma))$;
3. $f_\sigma := \Gamma(N_{good}, \sigma)$ for all $\sigma \in \Sigma_{\mathcal{A}}(x) \cap \Sigma_c$;
4. start Supcon$(P^y)$;
5.     for each $\underline{\mathbf{G}}^y$ w.r.t. $\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y$
6.         $(\forall\sigma \in \Sigma_{\mathcal{A}}(x) \cap \Sigma_c)\neg f_\sigma \wedge \underline{P_m}^y \wedge Elig_{\mathbf{G}}(\sigma) \models B$;
7.         if $B \neq \perp$
8.             $P^y = P^y \wedge \neg B$, $\underline{P_0}^y = \underline{P_0}^y \wedge C^x$, $\underline{P_m}^y = \underline{P_m}^y \wedge C^x$;
9.             $C^y = \sup\mathcal{C}^2\mathcal{P}(P^y)$;
10.            $N_{good} := \Theta(Next_{\mathbf{G}}(\sigma)) \wedge C^x$ for all $\sigma \in \Sigma_{\mathcal{A}}(y) \cap \Sigma_c$;
11.            $f_\sigma := \Gamma(N_{good}, \sigma) \vee \neg C^x$;
12.        endif
13.        if $C^y = \perp$ or $\underline{P_m}^y \wedge P^y \wedge \neg R(C^y) \neq \perp$,
14.         pause;//The structure of $\underline{\mathbf{G}}^y$ needs remodel.
15.        else
16.            for each $\underline{\mathbf{G}}^z$ w.r.t. $\underline{\mathbf{G}}^y < \underline{\mathbf{G}}^z$
17.                Supcon$(P^z)$;
18.            endfor
19.        endif
20.    endfor
21. end
22. return $f_\sigma$ for all $\sigma$ in $\Sigma_c$;
```

Suppose that $\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y$. As shown in Fig. 6, in each STS nest $\underline{\mathbf{G}}^y$, the system behavior is recorded in an agent $\underline{\mathbf{G}}^y_{traker}$. With respect to the specification for $\underline{\mathbf{G}}^y$, according to the optimal behavior C^x of $\underline{\mathbf{G}}^x$ and the current status (a basic state-tree b) provided by $\underline{\mathbf{G}}^y_{traker}$, a set of decision makers f_{σ_i}, with $\sigma_i \in \Sigma_c \cap \Sigma_{\mathcal{A}}(y)$ and $i = 1, 2, \ldots, n$, makes the decisions applying b as the argument. If $f_{\sigma_i}(b) = 1$, then σ_i is allowed to occur. Otherwise, it is disabled.

4.2 Boundary Consistency of Supervisory Control

As stated in Section 1, a well-formed STS satisfies boundary consistency. In this study, this feature is extended to the supervisory control of STS.

Property 1. We say that an STS satisfies the *boundary consistency of supervisory control* if it satisfies: the low level closed-loop (under control) STS nests can be "plugged" into the states of a high level STS nest without changing its control functions (control logics).

[1] Let $\mathbf{ST} = (X, x_0, \mathcal{T}, \mathcal{E})$ be a state-tree and $subST = (Y, x_0, \mathcal{T}', \mathcal{E}')$ be a sub-state-tree of $\mathbf{ST}$. Let $z \in Y$ and $\mathcal{T}'(z) = OR$. We say $x \in \mathcal{E}'(z)$ is *active* if $\mathcal{E}'^*(x) = \mathcal{E}^*(x)$ & $\mathcal{E}'^*(z) \subset \mathcal{E}^*(z)$, i.e., x is *active* if all of its descendants on $\mathbf{ST}$ are on $subST$ but at least one descendant of z is not on the $subST$. Each proper sub-state-tree $T \in \mathcal{ST}(\mathbf{ST})$ corresponds to an unique *active state set* $\mathcal{V}(T) = \bigcup_{\forall z \in Z} \mathcal{V}(z)$, where Z is the set of OR superstates that have *active* children.

[2] In the BDD representation, the OR (resp., SIM) states are considered as *variables* (resp., *values*). According to Ma & Wonham (2005, 2006), a state-tree $\mathbf{ST}$ is encoded into a predicate P by function $\Theta : \mathcal{ST}(\mathbf{ST}) \to Pred(\mathbf{ST})$. Let $\mathbf{ST}_1 = (X_1, x_{1,0}, \mathcal{T}_1, \mathcal{E}_1)$ be a sub-state-tree of $\mathbf{ST}$. Function $\Theta : \mathcal{ST}(\mathbf{ST}) \to Pred(\mathbf{ST})$ is recursively defined by

$$\Theta(\mathbf{ST}_1) = \begin{cases} \bigwedge\limits_{y \in \mathcal{E}_1(x_0)} \Theta'(\mathbf{ST}_1^y), & \text{if } \mathcal{T}(x_0) = AND \\ \bigvee\limits_{y \in \mathcal{E}_1(x_0)}, ((v_{x_0} = y) \wedge \Theta'(\mathbf{ST}_1^y)) & \text{if } \mathcal{T}(x_0) = OR \\ 1, & \text{if } \mathcal{T}(x_0) = SIM \end{cases}$$

where $\mathbf{ST}_1^y$ is the child-state-tree of $\mathbf{ST}_1$ rooted by y, and assume that $\Theta' : \mathcal{ST}(\mathbf{ST}) \to Pred(\mathbf{ST})$ is already defined on the child-state-tree $\mathbf{ST}_1^y$. Trivially, define $\Theta(\mathbf{ST}_1) \equiv 0$ if $\mathbf{ST}_1$ is an empty state-tree.

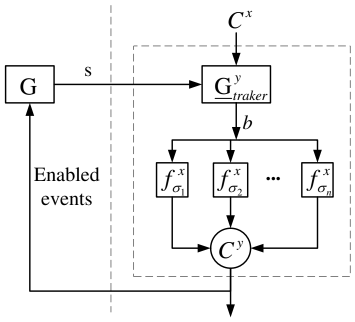

Fig. 6. Nested STS control diagram.

As the two-level predicate depicted in Fig. 7, suppose that event σ is controllable and it is disabled at basic-state-tree 2 on the top level. Within predicate 2 on the lower level, in order to avoid blocking the system at basic-state-tree c, it is considered as a new illegal basic-state-tree. Line 8 in Algorithm 2 guarantees that Property 1 is satisfied. Finally, after calculating the supervisor for the lower-level predicate 2, event τ is disabled.

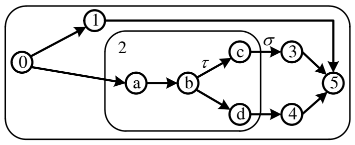

Fig. 7. Two level predicates.

5. CASE STUDIES

Two case studies are presented in this section to demonstrate the nested supervisory control of STS.

5.1 Transfer Line

For the transfer line studied in Section 3.1, the nonblocking supervisory control functions are:

- In $\underline{\mathbf{G}}^{TL}$:
  - · event 1 is enabled at: $\{B1_0,\ B2_0,\ M2_0,\ TU_0\}$,
  - · event 3 is enabled at: $\{B1_1\}$, and
  - · event 5 is enabled at: $\{B2_1\}$;
- In $\underline{\mathbf{G}}^{M1_1}$: events 11 and 13 are always enabled; and
- In $\underline{\mathbf{G}}^{M2_1}$: events 21 and 23 are always enabled.

The transfer line satisfies Property 1. The system behavior of its STS model under nested supervisory control is shown in Fig. 8. By projecting out the low-level behavior shown in Fig. 8 in the dashed line boxes, we obtain a diagram identical with the optimal behavior of the top level.

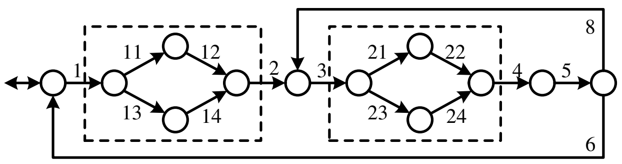

Fig. 8. Optimal behavior of Transfer Line.

5.2 AIP Example

The diagram of the AIP studied in Ma & Wonham (2005, 2006) is depicted in Fig. 9. AIP has five conveyor loops: one central loop communicates with four external loop by four transfer units. Linked to the external loops are three assembly stations and an I/O station. The primary DES model of AIP studied in Brandin (1994) is the synchronous product of 100 automata with a state space up to 10^{24}. Based on the developed nested SFBC, we obtain 36 different STS nests on three hierarchical levels. As a consequence, the total state space of all the 36 STS nests are around 2×10^{18}. The computation is finished in several seconds on a personal computer with 2.40 GHz Intel CPU and 8G RAM. The BDD nodes of the local control functions for several important controllable events are listed in Table 1 to compare between the the AIP studied in Ma & Wonham (2005, 2006) (under MW) and this study, in which the BDD size 0 represents that the corresponding event is allowed to occur when it is eligible.

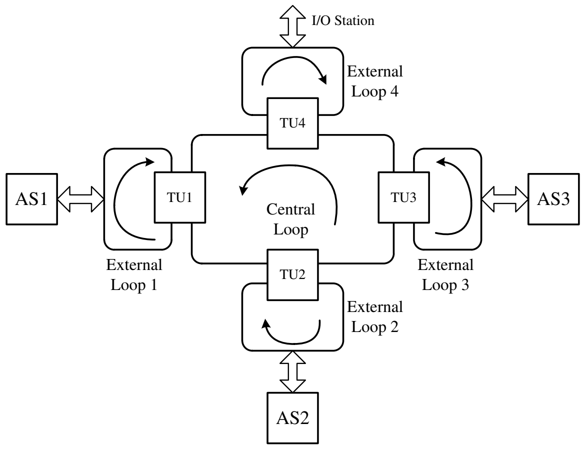

Fig. 9. AIP diagram.

Table 1. BDD size of Controller functions for AIP

Event	MW	This study
ASi_repaired ($i = 1, 2$)	0	0
ASi_stop_close ($i = 1, 2$)	1	9
ASi_stop_open ($i = 1, 2$)	16	9
ASi_gate_open ($i = 1, 2$)	0	0
ASi_read ($i = 1, 2$)	0	0
AS1_pickup3	15	0
AS1_pickup4	15	0
AS3_gate_open	0	0
AS3_read	2	2
L1_gate_open	95	44
CL_TUi_gate_open ($i = 1, 2$)	70	37
CL_TU1_stop_close	28	20
CL_TU2_stop_close	36	19
TUi_Drw2Li($i = 1, 2$)	54	0

6. DISCUSSIONS

For the one-level transfer line depicted in Fig. 4, by following the approach proposed in Ma & Wonham (2005, 2006) (under MW) and this study, the control functions for events 1, 3, and 5 are identical, as given in Section 5.1 under $\underline{\mathbf{G}}^{TL}$. However, after plugging the (nonblocking) holons shown in Fig. 5, their results are different. The BDD nodes of the control functions are listed in Table 2, which shows that Property 1 is not satisfied in the supervisory control of STS proposed in Ma & Wonham (2005, 2006).

Table 2. BDD nodes of controllers

Event	MW	This study
1	0	4
3	1	1
5	5	1
11	4	0
13	4	0
21	2	0
23	2	0

More precisely, by following Ma & Wonham (2005, 2006), control functions f_{11} and f_{13} make decisions based on the system behavior in some high level holons. They require that events 11 and 13 should occur if

- buffers $B1$ and $B2$ and test unit TU are empty, or
- buffer $B1$ is empty, buffer $B2$ is occupied, machine $M2$ is at the initial state, and test unit TU is empty.

As a consequence, the control functions calculated based on the approach proposed in Ma & Wonham (2005, 2006) may contain redundant control logics. This is caused by the redundant calculation of the synchronous product of $\underline{\mathbf{G}}^{M1_1}$ and $\underline{\mathbf{G}}^{M2_1}$. By following Ma & Wonham (2005) and Ma & Wonham (2006), the closed-loop behavior of the STS contains 56 basic state-trees and 126 transitions. However, as shown in Fig. 8, according to the nested approach presented in this study, we obtain the closed-loop behavior of the STS contains 12 basic state-trees and 15 transitions.

7. CONCLUSION

Based on the *AND* and *OR* superstates of an STS, we formally decompose it into a set of STS nests that describes its system behavior on each hierarchical level. The subordination relation among different STS nests is also defined. Suppose that an STS nest $\underline{\mathbf{G}}^y$ is subordinated to another STS nest $\underline{\mathbf{G}}^x$. In $\underline{\mathbf{G}}^x$, the complex internal behavior of $\underline{\mathbf{G}}^y$ is represented by a simple state y. Instead of calculating the optimal supervisor of an STS monolithically, a top-down nested approach is presented in this study to calculate the optimal behavior of each STS nest. By avoiding the redundant calculation of the synchronous product in independent STS nests, the state spaces of both the system model and the supervisor are reduced significantly. Finally, two case studies are presented in Section 5 to demonstrate the nested supervisory control of STS. For the STS model of the AIP studied in Ma & Wonham (2005, 2006); Brandin (1994), it was originally with a state space up to 10^{24}. In this study, it is decomposed into 36 different STS nests on three hierarchical levels. As a result, the total state space of all the 36 STS nests is reduced to around 2×10^{18}. The supervisors for each STS nest satisfies the boundary consistency of supervisory control, i.e., the low level closed-loop (under control) STS nests can be "plugged" into the states of a high level STS nest without changing its control functions (control logics). In our future work, we will work on the nested supervisory control of state-tree structures with partial observations.

REFERENCES

H. Marchand and B. Gaudin. Supervisory control problems of hierarchical finite state machines. *Proc. 41st IEEE Conf. on Dec. and Cont.*, 1199–1204, 2002.

B. Gaudin and H. Marchand. Supervisory control of product and hierarchical discrete event systems. *European Journal of Control*, 10(2):131–145, 2004.

B. Gaudin and H. Marchand. Safety control of hierarchical synchronous discrete event systems: A state-based approach. *Proc. IEEE Intern. Symp., Medit. Conf. Cont. Autom. Intel. Cont.*, 889–895, 2005.

D. Harel and A. Pnueli. On the development of reactive systems. In Logics and Models of Concurrent Systems. *NATO ASI Series*, 13:477–498, New York, 1985.

D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.

P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Contr. Optim.*, 25(1):206–230, 1987.

W. M. Wonham and K. Cai. *Supervisory control of discrete-event systems*, Monograph Series Communications and Control Engineering, Springer, 2018.

R. Alur, S. Kannan, M. Yannakakis. Communicating hierarchical state machines. *International Colloquium on Automata, Languages, and Programming*, 169–178, 1999.

C. Ma and W. M. Wonham. *Nonblocking Supervisory Control of State Tree Structures*, vol. 317, LNCIS, Berlin: Springer-Verlag, 2005.

C. Ma and W. M. Wonham. Nonblocking supervisory control of state tree structures. *IEEE Trans. Autom. Cont.*, 51(5):782–793, 2006.

R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, 35(8):677–691, 1986.

W. J. Chao, Y. M. Gan, Z. A. Wang, and W. M. Wonham. Modular supervisory control and coordination of state tree structures. *Intern. J. Cont.*, 86(1):9–21, 2013.

K. Cai and W. M. Wonham. *Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems*, vol. 459, LNCIS, Berlin: Springer-Verlag, 2015.

T. Jiao, Y. M. Gan, G. C. Xiao, and W. M. Wonham. Exploiting symmetry of state tree structures for discrete-event systems with parallel components. *Intern. J. Cont.*, 90(8):1639–1651, 2017.

C. Gu, X. Wang, Z. W. Li, and N. Q. Wu. Supervisory control of state-tree structures with partial observation. *Inform. Sci.*, Elsevier, 465(8):523–544, 2018.

C. Gu, X. Wang, and Z. W. Li. Synthesis of supervisory control with partial observation on normal state-tree structures. *IEEE Trans. Autom. Sci. Eng.*, 16(2):984–997, 2019.

D. G. Wang, X. Wang, and Z. W. Li. Nonblocking supervisory control of state-tree structures with conditional-preemption matrices. *IEEE Trans. Ind. Inform.*, 16(6):3744–3756, 2020.

B. Brandin and F. Charbonnier. The supervisory control of the automated manufacturing system of the AIP. *Proc. Rensselaer's 4th Int. Conf. on Computer Integrated Manufacturing and Automation Technology*, Troy, NY, 1994, 319–324.