Supervisory Control of Extended Finite Automata Using Transition Projection

Supervisory Control of Extended Finite Automata Using Transition Projection

Mohammad Reza Shoaei, Lei Feng, Bengt Lennartson

***Abstract*—A limitation of the Ramadge and Wonham (RW) framework for the supervisory control theory is the explicit state representation using finite automata, often resulting in complex and unintelligible models. Extended finite automata (EFAs), i.e., deterministic finite automata extended with variables, provide compact state representation and then make the control logic transparent through logic expressions of the variables. A challenge with this new control framework is to exploit the rich control structure established in RW's framework. This paper studies the decentralized control structure with EFAs. To reduce the computational complexity, the controller is synthesized based on model abstraction of subsystems, which means that the global model of the entire system is unnecessary. Sufficient conditions are presented to that guarantee the decentralized supervisors result in maximally permissive and nonblocking control to the entire system.**

I. Introduction

Supervisory control theory (SCT), established by Ramadge and Wonham [1], is a formal framework for the modeling and control of *discrete-event systems* (DES). Problems that SCT can address include dynamic resource allocation, system blocking prevention, etc. and, within these constraints, maximally permissive system behavior. Although SCT can systematically synthesize supervisory controllers that are able to prevent a DES from executing undesirable behavior, industrial acceptance is scarce. A number of issues that hinder industrial use have been identified by various researchers such as [2], [3]. Two main issues are the lack of a compact representation of large models and computational complexity.

In the former case, *Extended Finite Automata* (also called *Symbolic Transition Systems*), which are ordinary automata augmented with discrete variables, guard expressions and action functions, are introduced in [4] and [5]. Extended Finite Automata (EFAs) have been used in several research works and successfully applied to a range of examples such as [6], [7]. Beside a number of methods for synthesizing EFAs [8], [9], [10], the EFA framework in [5] has been implemented in Supremica [11], a verification and supervisory control tool.

Even though EFAs simplify the modeling experience by providing a compact modeling and representation, SCT analysis is still performed on their underlying automata models and therefore, the fundamental obstruction to the development of SCT, i.e., the computational complexity of synthesizing optimal nonblocking supervisors, still remains. Indeed, the nonblocking supervisory control problem for DES is NP-hard [12]. It is well known that the exponential complexity of supervisor design arises from synchronizing subsystems into a global system. Researchers are seeking effective control methods for various subclasses of DES that enjoy special structures. Such structures will admit modularity [13], [14], [15], [16] and model abstraction [17], [18], [19], [20] to circumvent computing global dynamic models.

The most effective model abstraction operator in SCT is the causal reporter map having the observer property [13]. While [21] treats hierarchical control using general causal reporter maps, Feng and Wonham [22], [23], construct model abstractions only with natural observers, i.e., natural projections [24], [25], [26] with the observer property. In this method, if two components share only a small number of common events, their abstractions tend to be small, and either verifying the nonconflicting property (if it holds) or designing a coordinator to achieve it may require only modest effort. Natural projection is a language-theoretic operation, which needs the language of a system to be known or can be obtained by its generators, for instance, automata. Unfortunately, this cannot be applied for DES modeled by EFAs. In particular, it makes no sense to speak of the language of individual extended automata, i.e., the language of the components can both be larger than or smaller than the language of the synchronized system. Hence, one cannot enjoy the compositional computation of natural projections.

In this paper, we introduce the *transition projection*, which is an extension of natural projection that can be applied directly on the transition systems of the EFAs, rather than their underlying finite automata. Sufficient conditions are presented for maximally permissive nonblocking and controllable controllers with partial observation in EFAs, by preserving the information needed for reliable representation of the nonblocking and controllability properties.

This paper is organized as follows: Section II briefly describes Extended Finite Automata modeling formalism used to model our problems. In Section III, we introduce a model abstraction using transition projection, that is the projection on transition systems, followed by Sections IV and V in which transition projection properties are explained. A practical example has been modeled and abstracted in Section VI and we conclude our work in Section VII.

M. R. Shoaei and B. Lennartson are with Department of Signals and Systems, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden, {shoaei, bengt.lennartson}@chalmers.se.

L. Feng is with the Department of Machine Design, KTH - Royal Institute of Technology, SE-100 44 Stockholm, Sweden, leifeng@md.kth.se.

This work was carried out at the Wingquist Laboratory VINN Excellence Center within the Area of Advance – Production at Chalmers, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). The support is gratefully acknowledged.

II. PRELIMINARIES

A. Languages and Automata

The behavior of DES [25], [26] is described in terms of event sequences and regular languages [1]. A regular language is a subset of strings that can be recognized by a *finite automaton* (FA) $G = (Q, \Sigma, \mapsto, q^0, Q^m)$. Q is the finite *state* set. $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$ is a non-empty finite event set called *alphabet*. $\mapsto \subseteq Q \times \Sigma \times Q$ is the state *transition relation* mapping elements of $Q \times \Sigma$ into singletons of Q. The element $q^0 \in Q$ is the *initial state* and $Q^m \subseteq Q$ is the set of *marked states*. The transition relation in G is written in infix notation $p \overset{\sigma}{\mapsto} q$. Let Σ^* be the set of all finite strings over Σ, including the empty string ε. Then, these notations can be extended to strings in Σ^* in the natural way by letting $p \overset{\varepsilon}{\mapsto} p$ for all $p \in Q$ and $p \overset{s\sigma}{\mapsto} q$ if $p \overset{s}{\mapsto} r$ and $r \overset{\sigma}{\mapsto} q$ for $s \in \Sigma^*, \sigma \in \Sigma, r \in Q$. Let $p \overset{\sigma}{\mapsto}$ denote the existence of at least one state q such that $p \overset{\sigma}{\mapsto} q$, and $p \mapsto q$ the existence of a string $s \in \Sigma^*$ such that $p \overset{s}{\mapsto} q$. Automaton G is deterministic if $p \overset{\sigma}{\mapsto} q$ and $p \overset{\sigma}{\mapsto} \acute{q}$ always implies $q = \acute{q}$. An important property of an automaton is *nonblocking*. The automaton G is nonblocking if any state reachable from the initial state q_0 can also reach a marked state via some string, i.e., $(\forall q \in Q)\ q_0 \mapsto q \Rightarrow q \mapsto p$ for some $p \in Q^m$.

Note that, by definition, the symbol ε does not belong to either of Σ, Σ_c, or Σ_u. If it is to be included, the event sets $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}, \Sigma_{\varepsilon,c} = \Sigma_c \cup \{\varepsilon\}$, and $\Sigma_{\varepsilon,u} = \Sigma_u \cup \{\varepsilon\}$ are used instead. Given two event sets Σ and $\Sigma_0 \subseteq \Sigma$, the *natural projection* is the function $P : \Sigma^* \to \Sigma_0^*$ such that $P(\varepsilon) = \varepsilon$,

$$P(\sigma) = \begin{cases} \varepsilon, & \sigma \in \Sigma - \Sigma_0 \\ \sigma, & \sigma \in \Sigma_0 \end{cases}$$
$$P(s\sigma) = P(s)P(\sigma), s \in \Sigma^*, \sigma \in \Sigma$$

The effect of P on a string $s \in \Sigma^*$ is just to erase the events in s that do not belong to Σ_0, but keep the events in Σ_0 unchanged. The *inverse image* of the natural projection P is a function $P^{-1} : Pwr(\Sigma_0^*) \to Pwr(\Sigma^*)$ where Pwr is the power set.

B. Extended Finite Automata

A finite automaton can be extended with a set of variables to an *Extended Finite Automaton* (EFA) whose transitions are augmented with conditions and actions on these variables to enjoy a compact and symbolic description of DES.

Let $V = \{v_1, \ldots, v_n\}$ be the set of n typed variables over the finite domain (type) $D = D_1 \times \cdots \times D_n$. Let $H(V)$ denote the set of (variable) evaluations η that assigns values to variables. $\mathcal{G}$ is the set of Boolean conditions over V in which each condition g, also called guard, is a propositional logic formula whose propositional symbols are of the form $\bar{v} \in \bar{D}$, where $\bar{v} = (v_1, \cdots, v_n)$ is an n-tuple of pairwise distinct variables in V, and $\bar{D}$ is a subset of the domain D. For the sake of simplicity, we write the propositional symbols such as "$x - y \leq 2$" instead of "$(x, y) \in \{(m, n) \in D_m \times D_n | m - n \leq 2\}$". A *satisfaction relation* $\models$ for a guard g can be defined as a set of pairs (η, g) indicating the evaluations η for which the guard g is satisfied. It is written $\eta \models g$ instead of $(\eta, g) \in \models$. Given two guards g and h, we say that g is a subguard of h, denoted $g \preceq h$, if $g \wedge h = g$. We say g and h have the same satisfaction, denoted $g = h$, if for all $\eta \in E(V)$, $\eta \models g$ iff $\eta \models h$.

Let $\mathcal{A}$ be the set of actions where each action $a \in \mathcal{A}$ is an n-tuple of partial function $a_i : D_i \to D_i$, updating the current variables value to a new value. The symbol ξ is used to indicate that a variable is not updated, namely, takes their current values.

Definition 1 (Extended Finite Automaton)**.**
An extended finite automaton over the set of variables V is a tuple $(L, D, \Sigma, T, \ell^0, d^0, L^m, D^m)$ where

- L is a finite set of discrete locations,
- $D = D_1 \times \cdots \times D_n$ is the finite domain of variables,
- Σ is a nonempty finite set of events (alphabets),
- $T \subseteq L \times \Sigma \times \mathcal{G} \times \mathcal{A} \times L$ is the transition relation,
- $\ell^0 \in L$ is the initial location,
- $d^0 \in D$ is the tuple of initial values,
- $L^m \subseteq L$ is the set of marked (desired) locations,
- $D^m \subseteq D$ is the set of marked values of the variables.

The notation $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$ is used as shorthand for $(\ell, \sigma, g, a, \acute{\ell}) \in T$. If the guard of the conditional transition $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$ is a tautology then we simply write $\ell \overset{\sigma}{\to}_{a} \acute{\ell}$. For two EFAs E_1 and E_2, with the same set of events, domain, initial location, and initial variables value we say E_1 is a *sub-EFA* of E_2, written $E_1 \subseteq E_2$, if $L_1 \subseteq L_2, T_1 \subseteq T_2, D_1^m \subseteq D_2^m$, and $L_1^m \subseteq L_2^m$.

The semantics of an EFA is described in terms of a DFA.

Definition 2 (EFA Semantics)**.**
Let $E = (L, D, \Sigma, T, \ell^0, d^0, L^m, D^m)$ be an EFA over the set of variables V. The DFA $G(E)$ of E is the tuple $(Q_E, \Sigma_E, \mapsto_E, q_E^0, Q_E^m)$ where $Q_E = L \times D$, $\Sigma_E = \Sigma$, $q_E^0 = \langle \ell^0, d^0 \rangle$, $Q_E^m = L^m \times D^m$, $\mapsto_E \subseteq Q \times \Sigma \times Q$ is defined by the following rule:

$$\frac{\ell \overset{\sigma}{\to}_{g/a} \acute{\ell} \wedge \eta \models g}{\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \acute{\ell}, a(\eta) \rangle}.$$

States of $G(E)$ are the set of reachable states of E, and each state consists of a location ℓ together with an evaluation η. Note that in the definition of transition relation $\mapsto$, if the proposition above the horizontal line holds, then the proposition under the line holds as well, namely, whenever the guard g of the conditional transition $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$ holds for the evaluation η, i.e., $\eta \models g$, then there is a transition in $G(E)$ from state $\langle \ell, \eta \rangle$ to state $\langle \acute{\ell}, a(\eta) \rangle$. Also, the DFA generated directly from a given EFA by constructing the state set as $L \times D$ is not guaranteed to be the canonical recognizer and therefore further reduction needs to be done by using the standard algorithm of minimization [27]. In the sequel, we assume that the DFA obtained by the above transformation is a canonical recognizer of the language represented by the input EFA model.

EFAs similar to ordinary finite automata are composed by extended full synchronous composition (EFSC). By the

definition of EFSC, it is assumed that the variables are shared by all EFAs with the same initial values.

Definition 3 (EFSC)**.**
Let $E = (L_k, D, \Sigma_k, T_k, \ell_k^0, d^0, L_k^m, D^m),\ k = 1, 2$, be two EFAs over the set of shared variables V. The *Extended Full Synchronous Composition* of E_1 and E_2 is the tuple

$$E_1 \| E_2 = (L, D, \Sigma, T, \ell^0, d^0, L^m, D^m),$$

where $L = L_1 \times L_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $\ell^0 = \langle \ell_1^0, \ell_2^0 \rangle$, $L^m = L_1^m \times L_2^m$, and T is defined by the following rules:

$$* \ \frac{\ell_1 \xrightarrow{\sigma}_{1,g_1/a_1} \acute{\ell}_1 \wedge \ell_2 = \acute{\ell}_2 \wedge \sigma \in (\Sigma_1 - \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma}_{g_1/a_1} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle};$$

$$* \ \frac{\ell_2 \xrightarrow{\sigma}_{2,g_2/a_2} \acute{\ell}_2 \wedge \ell_1 = \acute{\ell}_1 \wedge \sigma \in (\Sigma_2 - \Sigma_1)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma}_{g_2/a_2} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle};$$

$$* \ \frac{\ell_1 \xrightarrow{\sigma}_{1,g_1/a_1} \acute{\ell}_1 \wedge \ell_2 \xrightarrow{\sigma}_{2,g_2/a_2} \acute{\ell}_2 \wedge \sigma \in (\Sigma_1 \cap \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \xrightarrow{\sigma}_{g/a} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$$

such that $g = g_1 \wedge g_2$ and for $i = 1, \ldots, n$ we have

$$a_i = \begin{cases} a_{1i} & \text{if } a_{1i} = a_{2i} \\ a_{1i} & \text{if } a_{2i} = \xi \\ a_{2i} & \text{if } a_{1i} = \xi \\ \eta_i & \text{otherwise;} \end{cases}$$

Note that, if the action functions of E_1 and E_2 try to update a shared variable to different values, the variable is, by default, not updated.

We introduce the notion of local events for a system consisting of more than one EFA component, which will be used later. For an event σ, let $Act(\sigma) \subseteq \mathcal{A}$ and $Con(\sigma) \subseteq \mathcal{G}$ be the sets of actions and guards, respectively, retrieved from all transitions labeled with σ.

Definition 4 (Local Event)**.**
An event $\sigma \in \Sigma_k$ is *local* to $E_k, k \in \Omega_E$ where Ω_E is an index set, if for all $m \in \Omega_E$ we have (i) $\sigma \in \Sigma_k - \bigcup \Sigma_m (k \neq m)$, (ii) $(\forall g \in Con(\sigma))$ g is a tautology, (iii) $(\forall a \in Act(\sigma); \forall \eta \in H(V); \forall g \in \bigcup \mathcal{G}_m)$ $\eta \models g \Leftrightarrow a(\eta) \models g$.

In above, condition (i) guarantees that the event σ only appears in E_k, (ii) ensures that guards on any transition labeled by σ evaluates to true; hence σ can cause the transition to occur at any time, and (iii) guarantees that the actions on all transitions labeled by σ have no effect on any guard in the system. We say that an event is *shared* with other EFAs when it is not local. Also, any transition labeled with a local event is called a *local transition*.

Moreover, we use the notion of *executions* (also called *runs*) to describe a possible behavior of the transition system.

Definition 5 (Execution Fragment)**.**
An *execution fragment* ϱ in E is a series of finite transitions in T, $\varrho = \ell_0 \xrightarrow{\sigma_1}_{g_1/a_1} \ell_1 \xrightarrow{\sigma_2}_{g_2/a_2} \cdots \xrightarrow{\sigma_{i+1}}_{g_{i+1}/a_{i+1}} \ell_{i+1},\ (0 \leqslant i < n)$, where $n \geqslant 0$ and the variables evaluation $\eta_{i+1} = a(\eta_i)$. The integer n is the length of the fragment ϱ and $\varrho = \ell_0$ for some $\ell_0 \in L$ is a legal execution fragment of length $n = 0$.

The first and last location of ϱ is denoted by *first*(ϱ) and *last*(ϱ), respectively. We call an execution fragment ϱ *initial* if *first*$(\varrho) = \ell^0$, *marked* if *last*$(\varrho) \in L^m$, and *local* if all of its transitions are local. For two execution fragments $\varrho, \acute{\varrho}$, we say ϱ is a *precedence* of $\acute{\varrho}$, written $\varrho \sqsubseteq \acute{\varrho}$, if *last*$(\varrho)$ = *first*$(\acute{\varrho})$ and we say $\varrho = \acute{\varrho}$ if they have the same sequence of transitions up to renaming of locations.

C. Supervisory Control of EFAs

SCT is a formal framework for the modeling and control of DES consisting of a plant and a specification. A supervisor for a control problem modeled by EFAs can be symbolically computed using the algorithm presented in [10]. The algorithm iteratively strengthens the guards on conditional transitions to avoid forbidden or blocking states.

Given a DES control problem, we assume that the plant is modeled by an EFA G and the specification by an EFA K. The specification can be represented, without loss of generality, by a set of forbidden locations, which can be obtained by a refined plant model R with the same behavior as G such that the executions not allowed in K end up in certain forbidden locations in R. See [10] for more elaboration on refinement. From now on, we assume that the plant model is given as the refined EFA R and the specification is given as the set of forbidden locations $L_f \subset L_R$. Let us denote the set of safe locations by $L_s = L - L_f$, and recall the set of reachable states Q_R in $G(R)$. A state $q = \langle \ell, \eta \rangle \in Q_R$ is a forbidden state iff $\ell \in L_f$, otherwise, q is a safe state. In the sequel, R_s denotes the EFA obtained from R by assigning *false* to the guard g of every transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$ for which $\acute{\ell} \in L_f$, i.e., $\acute{\ell}$ is a forbidden location. R_s is constructed such that $R_s \subseteq R$ and is called the safe sub-EFA of R.

Definition 6 (Nonblocking, Safety, Controllability)**.**
[10] Let R be an EFA, L_f its set of forbidden locations, and R_s its safe subautomaton. A reachable state $q \in Q_R$ is: (a) *nonblocking* if there exists a state $p \in Q_R^m$ such that $q \overset{s}{\mapsto} p$ for some string $s \in \Sigma^*$; (b) safe if $q \in Q_{R_s}$ and (c) (R, L_f, Σ_u)-controllable (or simply *controllable* when clear from context) if q is safe and $\forall \sigma \in \Sigma_R(q) \cap \Sigma_u$ where $\Sigma_R(q)$ denote the set of active events, we have $Q_R(q, \sigma) \subseteq Q_{R_s}$. The EFA R is, respectively, nonblocking, safe, and controllable if every reachable state of R is, respectively, nonblocking, safe, and controllable.

A supervisor $\mathcal{S}$ for R can be seen as a function $\mathcal{S} : T \to \mathcal{G}$ which maps each transition to a supervision guard such that $\mathcal{S}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) \preceq g$ if $\sigma \in \Sigma_c$, and $\mathcal{S}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) = g$ if $\sigma \in \Sigma_u$. Let $R^{\mathcal{S}}$ denote the sub-EFA obtained from R by replacing its guards by those provided by $\mathcal{S}$. Then, $\mathcal{S}$ is said to be nonblocking if $R^{\mathcal{S}}$ is nonblocking and safe if $R^{\mathcal{S}}$ is safe. In case $R^{\mathcal{S}}$ is blocking or uncontrollable, a search will be performed to find a safe and nonblocking supervisor $\mathcal{S}$ such that $R^{\mathcal{S}} \subseteq R_s$. Let $\mathcal{S}(R, L_f)$ denote the set of nonblocking and safe supervisor candidates of R, then $\mathcal{S}^{\uparrow} := sup\mathcal{S}(R, L_f)$, is the *most permissive nonblocking and safe supervisor* compared to any other supervisor in

$\mathcal{S}(R, L_f)$ when the latter is nonempty. The $R^{\mathcal{S}^\uparrow}$ is called the supremal controllable and nonblocking sub-EFA of R_s.

$R^{\mathcal{S}^\uparrow}$ is calculated by the Supervisory Synthesis for EFA (SSEFA) using a fixed-point iteration method. Given a refined EFA R and a set $L_f \subset L$ of forbidden location, SSEFA(R, L_f) computes stronger, maximally permissive, guards for the transitions of R in N steps such that the obtained EFA is nonblocking, safe and controllable [10].

III. EFA Projection

Traditionally, brute-force computation is used for verification and coordination [25], [17]. This we wish to avoid since the nonblocking supervisory control problem in SCT [17] is NP-hard [12]. Abstraction introduces hierarchy into the system structure, as it reports only the events shared with other subsystems and conceals the rest. The fewer the reported events, the greater state reduction will be achieved. In order to use a model abstraction using natural projection on EFAs, we introduce the *transition projection* which is an extension of natural projection to abstract systems modeled by EFAs. We present sufficient conditions for an optimal nonblocking and controllable supervisor with partial observation in EFA, by preserving the information needed for reliable representation of the nonblocking and controllability properties.

For an EFA E with the set of events Σ, the transition projection, written with a slight abuse of notation $\bar{P}$, for the conditional transition relation T and the set $\Sigma_\ell \subseteq \Sigma$ is a function $\bar{P} : T \times \Sigma_\ell \to T$ defined as follows: for every transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \in T$,

$$\bar{P}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \varepsilon) = \ell \xrightarrow{\sigma}_{g/a} \acute{\ell},$$

$$\bar{P}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \gamma) = \begin{cases} \ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, & \sigma \neq \gamma \\ \ell \xrightarrow{\varepsilon}_{g/a} \acute{\ell}, & \sigma = \gamma. \end{cases}$$

The transition projection $\bar{P}$ replaces the label of transitions labeled by events in Σ_ℓ with the symbol ε. In effect, an EFA is allowed to make a transition spontaneously, without receiving an input event. Extending T to its power set $Pwr(T)$, we get $\bar{P} : Pwr(T) \times \Sigma_\ell \to Pwr(T)$ such that for any $\tau \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, \tau) = \{\bar{P}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \tau) | \ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \in N\}$. If we further extend Σ_ℓ to its power set $Pwr(\Sigma_\ell)$, $\bar{P}$ becomes $\bar{P} : Pwr(T) \times Pwr(\Sigma_\ell) \to Pwr(T)$ such that for $A \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, A) = \bigcup\{\bar{P}(N, \tau) | \tau \in A\}$. If the action of $\bar{P}$ on T is understood then $\bar{P}(T, \Sigma_\ell)$ may be written $\bar{P}_{\Sigma_\ell} T$ and similarly if $\bar{P}$ is defined then $\bar{P}T$.

Given any EFA, Algorithm 1, denoted by $\hat{P}$, computes the projected EFA. The intuition of the algorithm is the following. Let $\mathcal{S}_\varepsilon(\ell)$ be the set of ε-closure of a location ℓ in E. $\mathcal{S}_\varepsilon(\ell)$ is constructed recursively by finding every location that can be reached from ℓ along any path whose transitions are all labeled ε. Formally, (1) $\ell \in \mathcal{S}_\varepsilon(\ell)$, (2) $(\forall \acute{\ell} \in \mathcal{S}_\varepsilon(\ell))\ \acute{\ell} \xrightarrow{\varepsilon}_{g/a} \grave{\ell} \Rightarrow \grave{\ell} \in \mathcal{S}_\varepsilon(\ell)$. The location set of $\tilde{E}$ will be denoted by $\tilde{L}$, with element $\tilde{\ell}$ that label ε-closure subsets of E. The transition system of $\tilde{E}$ is constructed as follows. Define the initial location subset $\tilde{\ell}^0 := \mathcal{S}_\varepsilon(\ell^0)$. Choose $\sigma_1 \in \Sigma - \Sigma_\ell$ and define $\tilde{\ell}^1 := \bigcup_{\ell \in \tilde{\ell}^0}\{\mathcal{S}_\varepsilon(\acute{\ell}) \mid (\ell, \sigma_1, g, a, \acute{\ell}) \in T\}$. Define $\tilde{\ell}^2$ similarly, from $\tilde{\ell}^0$ and $\sigma_2 \in \Sigma - \Sigma_\ell - \{\sigma_1\}$, and repeat until $\Sigma - \Sigma_\ell$ is exhausted. The subset obtained at any step is discarded if it is empty or if it appeared previously. This process yields a list of (final) distinct nonempty subsets $\tilde{\ell}^0, \tilde{\ell}^1, \ldots, \tilde{\ell}^{k_1}$ and one-step 'subset' transitions of form $(\tilde{\ell}^0, \sigma, g, a, \tilde{\ell}^i)$, $\sigma \in \Sigma - \Sigma_\ell$, $i \in \{0, 1, \ldots, k_1\}$. The procedure is repeated with each of the subsets $\tilde{\ell}^1, \tilde{\ell}^2, \ldots, \tilde{\ell}^{k_1}$ and each $\sigma \in \Sigma - \Sigma_\ell$, until no new subset transitions are obtained. The result is the projected EFA $\tilde{E} = (\tilde{L}, D, \tilde{\Sigma}, \tilde{T}, \tilde{\ell}^0, d^0, \tilde{L}^m, D^m)$, where $\tilde{L}$ is the final list $\{\tilde{\ell}^0, \tilde{\ell}^1, \ldots, \tilde{\ell}^k\}$, $\tilde{L}^m := \{\tilde{\ell} \in \tilde{L} \mid \tilde{\ell} \cap L^m \neq \emptyset\}$, and $(\tilde{\ell}, \sigma, g_\sigma, a_\sigma, \tilde{\ell}') \in \tilde{T}$ iff $(\ell, \sigma, g, a_\sigma, \acute{\ell}) \in T$ for some $\ell \in \tilde{\ell}, \acute{\ell} \in \tilde{\ell}', \sigma \in \Sigma - \Sigma_\ell$.

Algorithm 1 EFA Projection ($\hat{P}$)

Input: An EFA $E = (L, D, \Sigma, T, \ell^0, d^0, L^m, D^m)$ and a subset of events $\Sigma_\ell \subseteq \Sigma$.

```
Apply P̄ : T × Σ_ℓ → T to E;
Σ̃ := Σ − Σ_ℓ;
ℓ̃^0 := S_ε(ℓ^0);
L̃ := {ℓ̃^0};
S := {ℓ̃^0};
repeat
    X = ∅;
    for all ℓ̃^1 ∈ S and σ ∈ Σ̃ do
        ℓ̃^2 := ⋃_{ℓ∈ℓ̃^1} {S_ε(ℓ́) | (ℓ, σ, g, a, ℓ́) ∈ T};
        if ℓ̃^2 ≠ ∅ then
            if ℓ̃^2 ∉ L̃ then
                X := X ∪ ℓ̃^2;
                L̃ := L̃ ∪ ℓ̃^2;
            end if
            T̃ := T̃ ∪ {(ℓ̃^1, σ, g, a, ℓ̃^2)};
        end if
    end for
    S := X;
until S = ∅
L̃^m := {ℓ̃ ∈ L̃ | ℓ̃ ∩ L^m ≠ ∅};
```

Output: An EFA $\tilde{E} = (\tilde{L}, D, \tilde{\Sigma}, \tilde{T}, \tilde{\ell}^0, d^0, \tilde{L}^m, D^m)$.

We assume a DES to consist of a group of simple plant EFAs subject to a conjunction of modular control specifications. Consider a system consisting of two EFA components, E_1 and E_2. To obtain a reduction of the system, we could first compute the systems global behavior $E_1 \| E_2$ and then its transition projection. When, however, the local events of the two components are all defined the result is obtained more economically from reductions of the components.

Proposition 1.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, d^0, L_k^m, D^m), k = 1, 2$, be two EFAs. Consider T as the set of transition relation for $E_1 \| E_2$ and $\Sigma_\ell \subseteq \Sigma := \Sigma_1 \cup \Sigma_2$. Define $\bar{P} : T \times \Sigma_\ell \to T$ and $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i (i = 1, 2)$. If Σ_ℓ is the set of local events then $\hat{P}(E_1 \| E_2, \Sigma_\ell) = \hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell) \| \hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell)$.

Proof: See [28]. ∎

The extension to an arbitrary number of synchronized factors is straightforward and is left out.

IV. Observer of EFA

Consider a DES described by EFA E. Given a set of local events, we can define the transition projection $\bar{P}: T \times \Sigma_\ell \to T$ and then the projected EFA $\hat{P}(E, \Sigma_\ell)$. The resulting projected EFA is not guaranteed to be coreachable, or nonblocking for E. Crucial to successful model abstraction using transition projection is that the projected system contains necessary and sufficient information needed for reliable representation of the nonblocking property. Therefore, one must carefully select the local events of a DES.

A "good" selection of local events for any transition projection is whenever a projected EFA reaches a marked location via some projected execution fragments, the original system must be able to reach a marked location by those execution fragments as follows.

Definition 7 (E-observer)**.**
Assume a nonblocking EFA E and let $\Sigma_\ell \subseteq \Sigma$ be the subset of local events. The transition projection $\bar{P}: T \times \Sigma_\ell \to T$ is an *E-observer*, if for all initial execution fragments ϱ_s and $\varrho_{\acute{s}}$ and for all marked execution fragment ϱ_t in E such that $\varrho_s \sqsubseteq \varrho_t$ and $\bar{P}\varrho_s = \bar{P}\varrho_{\acute{s}}$, there exists a marked execution fragment $\varrho_{\acute{t}}$ in E such that $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$ and $\bar{P}\varrho_{\acute{t}} = \bar{P}\varrho_t$.

Note that if Σ_ℓ is equal to Σ or $\emptyset$ for an EFA E then $\bar{P}$ is automatically an E-observer. It can be shown, by similar reasoning on reachable states as in [29], that the model abstractions computed by transition projection with observer property are guaranteed to have location sizes no larger than the original model.

For a system consisting of more than one plant component it would be more economical to check the observer property component-wise without computing the synchronous product first. Proposition 2 presents a sufficient condition for this simplification to be valid.

Proposition 2.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, d^0, L_k^m, D^m), k = 1, 2$, be two nonblocking EFAs. Consider T as the set of transition relation for $E_1 \| E_2$. Define the transition projections $\bar{P}: T \times \Sigma_\ell \to T$ and $\bar{Q}_i: T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1, 2)$ where $\Sigma_\ell \subset \Sigma := \Sigma_1 \cup \Sigma_2$. If Σ_ℓ is the set of local events and for both $i = 1, 2, \bar{Q}_i$ is an E_i-observer, then $\bar{P}$ is an $E_1 \| E_2$-observer.

Proof: See [28]. ∎

As we establish a "reliable interface" for EFAs by introducing E-observer, the interaction between two complex systems may be examined through their projections rather than their global behavior. If $\bar{P}$ has the observer property, we can check if two EFAs E_1 and E_2 are synchronously nonconflicting by checking whether their projections are synchronously nonconflicting and we may save significant computational effort, in accordance with the following.

Theorem 1 (Synchronously Nonconflicting Criterion)**.** *Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, d^0, L_k^m, D^m), k = 1, 2$, be two EFAs and let $\Sigma_\ell \subset \Sigma := \Sigma_1 \cup \Sigma_2$ be the set of local events. If $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ are E_i-observer $(i = 1, 2)$, then $E_1 \| E_2$ is nonblocking if and only if $\hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell) \| \hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell)$ is nonblocking.*

The proof needs the following Lemma. Denote $E = \emptyset$ when there is no outgoing transition from the initial location of E.

Lemma 1.
In the notation of Proposition 2, define the transition projection $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ $(i, j = 1, 2; j \neq i)$. If $\Sigma - \Sigma_\ell \neq \emptyset$ and there exists $\ell_i \xrightarrow{\sigma}_{g/a} \acute{\ell}_i \in T_i$ such that $\sigma \in \Sigma - \Sigma_\ell$ for some $\ell_i, \acute{\ell}_i \in L_i$ $(i = 1, 2)$ then $E_1 \| E_2 \neq \emptyset \Leftrightarrow \hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell) \| \hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell) \neq \emptyset$.

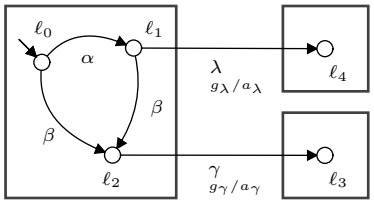
Proof: See [28]. ∎

Returning to the proof of Theorem 1, define the transition projections $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ $(j \neq i), \bar{Z} : T \times ((\Sigma_1 \cup \Sigma_2) - (\Sigma_1 \cap \Sigma_2)) \to T, \bar{R}_i := \bar{Q}_i \circ \bar{P}_i (i, j = 1, 2)$ and let $\tilde{E}_1 = \hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell), \tilde{E}_2 = \hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell)$.

Proof of Theorem 1: (**If**) Let ϱ_s be an initial execution fragment in $E_1 \| E_2$. We must show that there exists a marked execution fragment ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Apply $\bar{P}_i$ to ϱ_s, we get $\bar{P}_i\varrho_s \in E_i$ $(i = 1, 2)$. We also know that $\bar{P}(\varrho_s) \in \hat{P}(E_1 \| E_2, \Sigma_\ell)$. Because of the assumption that Σ_ℓ is the set of local events and by Proposition 1, $\bar{P}\varrho_s \in \tilde{E}_1 \| \tilde{E}_2$. Then, by Proposition 2 there must exist a marked execution fragment $\acute{\varrho}_t \in \tilde{E}_1 \| \tilde{E}_2$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$. Applying $\bar{R}_i$ on both sides, we get $\bar{R}_i\bar{P}\varrho_s$ and $\bar{R}_i\acute{\varrho}_t$.We have $\bar{R}_i \circ \bar{P} = \bar{Q}_i \circ \bar{P}_i$ $(i = 1, 2)$. Consequently, both $\bar{Q}_i\bar{P}_i\varrho_s$ and $\bar{R}_i\acute{\varrho}_t$ are in $\hat{Q}_i(E_i, \Sigma_i \cap \Sigma_\ell)(i = 1, 2)$. Since $\bar{P}_i\varrho_s \in E_i$ and $\bar{Q}_i$ is an E_i-observer, there exists a marked execution fragment $\varrho_{w_i} \in E_i$ such that $\bar{P}_i\varrho_s \sqsubseteq \varrho_{w_i}$ and $\bar{Q}_i\varrho_{w_i} = \bar{R}_i\acute{\varrho}_t$. Applying $\bar{P}_j (j = 1, 2; j \neq i)$ to both sides of this equation, we get $\bar{P}_j\bar{Q}_i\varrho_{w_i} = \bar{P}_j\bar{R}_i\acute{\varrho}_t = \bar{Z}\acute{\varrho}_t$ and $\bar{P}_j \circ \bar{Q}_i = \bar{P}_j$. This implies that $\bar{P}_2\varrho_{w_1} = \bar{Z}\acute{\varrho}_t = \bar{P}_1\varrho_{w_2}$. Constructing the set $\Pi := \{\varrho_w \in E_1 \| E_2 \mid \bar{P}_1\varrho_w = \varrho_{w_1} \wedge \bar{P}_2\varrho_w = \varrho_{w_2}\}$. We know that $\Pi \neq \emptyset$. Hence, taking any marked execution fragment form the set Π, say $\varrho_w \in \Pi$, we have $\bar{P}_i\varrho_w = \varrho_{w_i} (i = 1, 2)$. Since $\varrho_{w_i} \in E_i$, we have $\bar{P}_i\varrho_w \in E_i (i = 1, 2)$. Consequently, $\varrho_w \in E_1 \| E_2$, and as required is marked and $\varrho_s \sqsubseteq \varrho_w$.
(**Only if**) According to the assumption $E_1 \| E_2$ is nonblocking and therefore, for any initial execution fragment ϱ_s there exists a marked execution fragment ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Apply $\bar{P}$ on both ϱ_s and ϱ_t, we get, respectively, $\bar{P}\varrho_s$ and $\bar{P}\varrho_t$ in $\hat{P}(E_1 \| E_2, \Sigma_\ell)$, and by Proposition 1, they are also in $\tilde{E}_1 \| \tilde{E}_2$. Since $\bar{P}$ is $E_1 \| E_2$-observer, there must exist a marked execution fragment $\acute{\varrho}_t \in \tilde{E}$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ and $\bar{P}\acute{\varrho}_t = \bar{P}\varrho_t$. By Proposition 1, $\acute{\varrho}_t \in \tilde{E}_1 \| \tilde{E}_2$. Therefore, for any execution fragment $\bar{P}\varrho_s \in \tilde{E}_1 \| \tilde{E}_2$ there exists a marked execution fragment $\acute{\varrho}_t$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ which implies $\tilde{E}_1 \| \tilde{E}_2$ is also nonblocking. ∎

In case two EFAs E_1 and E_2 are synchronously conflicting, a third EFA E, called a coordinator, must be introduced to resolve the conflict. We can now, instead of computing the coordinator directly from the two EFAs themselves, perform this computation through their abstractions.

Proposition 3.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, d^0, L_k^m, D^m), k = 1, 2$, be two synchronously conflicting EFAs and let $\Sigma_\ell \subset \Sigma := \Sigma_1 \cup \Sigma_2$ be the set of local events. If $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ are



◯ are the locations of E and □ are the locations (subsets) of $\tilde{E}$

Fig. 1: For the EFA E with $\Sigma_\ell = \{\alpha, \beta\},\ \Sigma_u = \{\beta, \gamma\}$ the transition projection $\bar{P}$ is not OCC since there exists an execution fragment $\varrho = \ell_0 \xrightarrow{\alpha} \ell_1 \xrightarrow{\beta} \ell_2 \xrightarrow{\gamma}_{g_\gamma / a_\gamma} \ell_3$ such that $\gamma \notin \Sigma_\ell, \{\alpha, \beta\} \in \Sigma_\ell, \gamma \in \Sigma_u$ but $\alpha \notin \Sigma_u$.

E_i-observer $(i = 1, 2)$ and there exists an EFA E such that $\hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell) \| \hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell) \| E$ is nonblocking then $E_1 \| E_2 \| E$ is also nonblocking.

Proof: See [28]. ■

As long as E can resolve the conflict between $\hat{Q}_1(E_1, \Sigma_1 \cap \Sigma_\ell)$ and $\hat{Q}_2(E_2, \Sigma_2 \cap \Sigma_\ell)$, it can resolve the conflict between E_1 and E_2.

V. Optimal Nonblocking and Controllable Supervisor

An optimal supervisor with full observation usually disables the nearest controllable events preceding or "upstream" to a prohibited uncontrollable event (say, σ). If, however, some of these controllable events are unobservable, a decentralized supervisor must disable controllable events further back, and so is more restrictive. For this restriction to be relaxed, the local event set must be selected properly enough to contain all the upstream controllable events nearest to σ. Such a decentralized supervisor will prevent the occurrence of an uncontrollable event while allowing maximal freedom of system behavior. A transition projection with such a local event set is called Output Control Consistent (OCC).

Definition 8 (OCC)**.**
Let $E = (L, D, \Sigma, T, \ell^0, d^0, L^m, D^m)$ be an EFA and let $\Sigma_\ell, \Sigma_u \subseteq \Sigma$ be the local and uncontrollable event sets. The transition projection $\bar{P} : T \times \Sigma_\ell \to T$ is *output control consistent (OCC)* for the EFA E, if for every finite execution fragment ϱ of the form

$$\varrho = \ell_0 \xrightarrow{\sigma_1}_{g_1/a_1} \cdots \xrightarrow{\sigma_{i+1}}_{g_{i+1}/a_{i+1}} \ell_{i+1} \quad \text{or}$$
$$\varrho = \ell \xrightarrow{\sigma}_{g/a} \ell_0 \xrightarrow{\sigma_1}_{g_1/a_1} \cdots \xrightarrow{\sigma_{i+1}}_{g_{i+1}/a_{i+1}} \ell_{i+1},\ 0 \leqslant i < n$$

which satisfies the conditions that $n \geqslant 1, \sigma \in \Sigma - \Sigma_\ell, \sigma_j \in \Sigma_\ell\ (j \in \textbf{k-1})$ and $\sigma_n \in \Sigma - \Sigma_\ell$, we have the property that $\sigma_n \in \Sigma_u \Rightarrow (\forall j \in \mathbf{n})\ \sigma_j \in \Sigma_u$.

In above definition, when σ_n is not local and uncontrollable, its immediately preceding local events must all be uncontrollable, namely, its nearest controllable event must be observable.

Example 1. Consider EFA E in Fig. 1 where $\Sigma = \{\alpha, \beta, \lambda, \gamma\}, \Sigma_\ell = \{\alpha, \beta\}, \Sigma_u = \{\beta, \gamma\}$ are, respectively, the sets of alphabet, local events, and uncontrollable events. Let $\bar{P} : T \times \Sigma_\ell \to T$ be the transition projection. In this, $\bar{P}$ is not OCC for E since there exists an execution fragment $\varrho = \ell_0 \xrightarrow{\alpha} \ell_1 \xrightarrow{\beta} \ell_2 \xrightarrow{\gamma}_{g_\gamma / a_\gamma} \ell_3$ where $\gamma \in \Sigma - \Sigma_\ell, \alpha, \beta \in \Sigma_\ell$ but $\alpha \notin \Sigma_u$.

We can now state a sufficient condition for Optimal Nonblocking and Controllable Supervisor (ONCS).

Theorem 2 (ONCS)**.**
Let E be a nonblocking EFA along with local and uncontrollable event sets $\Sigma_\ell,\ \Sigma_u \subseteq \Sigma$, respectively. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and let the EFA $\tilde{E} = \hat{P}(E, \Sigma_\ell)$. Suppose the set of forbidden locations is $\tilde{L}_f \subset \tilde{L}$. If the transition projection $\bar{P}$ is an E-observer and OCC for E, then $supS(E, \tilde{L}_f) = supS(\tilde{E}, \tilde{L}_f) \| E$.

$supS(E, \tilde{L}_f)$ denotes the optimal nonblocking and controllable supervisor with full observation that can be obtained for E. Similarly, $supS(\tilde{E}, \tilde{L}_f)$ describes the decentralized supervisor with partial observation on $\Sigma - \Sigma_\ell$. When this supervisor is synchronized with the plant, the final controlled behavior is the $supS(\tilde{E}, \tilde{L}_f) \| E$.

Proof: It needs to be shown that the reachable states of the fixed points of $supS(E, \tilde{L}_f)$ and $supS(\tilde{E}, \tilde{L}_f) \| E$, i.e., $G^N := G(\text{SSEFA}(E, \tilde{L}_f)^N)$ and $\tilde{G}^N := G(\text{SSEFA}(\tilde{E}, \tilde{L}_f)^N \| E)$, respectively, are the same. This can be proved by an induction on the step iterator j.
$(\subseteq)$ **BASE:** Let $j = 0$. By definition $\tilde{G}^N \subseteq L \times D = G^0$.
INDUCTION: Assuming that the property holds for j it needs to be shown that it also holds for $j+1$. Let $p = \langle \ell, \eta \rangle \in \tilde{G}^N$. By the inductive assumption it holds that $p \in G^j$. Assume that $p \notin G^{j+1}$. This implies that either p is (α) uncontrollable or (β) blocking state and therefore removed by the synthesis algorithm.
(α) Then there exists $v \in \Sigma_u$ such that $p \overset{v}{\mapsto}_{G^j} q \notin G^j$ for some $q = \langle \acute{\ell}, \acute{\eta} \rangle \in L \times D$. Assume $v \in \Sigma - \Sigma_\ell$. Then v is not projected by $\bar{P}$ so the same transition exists in $\tilde{E}$. Therefore, $p \overset{v}{\mapsto}_{\tilde{G}^N} q \notin G^j \supseteq \tilde{G}^N$. But then $p \notin \tilde{G}^N$, which is a contradiction. Now, assume $v \in \Sigma_\ell$. Then for all execution fragments of the form $p \overset{\sigma_1}{\mapsto}_{G^j} q \overset{\sigma_2}{\mapsto}_{G^j} \cdots \overset{\sigma_{k-1}}{\mapsto}_{G^j} t \overset{\sigma_k}{\mapsto}_{G^j} u\ (k \geq 1)$ in G^j such that $\sigma_1 = v$ and satisfies the conditions $\sigma_1, \ldots, \sigma_i \in \Sigma_\ell (i \in \textbf{k-1})$ and $\sigma_k \in \Sigma - \Sigma_\ell$. Observe that the subset of states $\{p, \cdots, t\}$ is labeled by p in $\tilde{E}$. If $\sigma_k \in \Sigma_u$ then by the assumption that $\bar{P}$ is OCC for E we can immediately see $(\forall i \in \mathbf{k}) \sigma_i \in \Sigma_u$. Consequently, $p \overset{\sigma_1}{\mapsto}_{G^j} \cdots \overset{\sigma_k}{\mapsto}_{G^j} u \notin G^j \supseteq \tilde{G}^N$ and $p \notin \tilde{G}^N$ which is a contradiction. If $\sigma_k \in \Sigma_c$ then it must be the case that $q \overset{\sigma_2}{\mapsto}_{G^j} \notin G^j$. Assume $\sigma_2 \in \Sigma_c$ then $q \in G^j$ which is a contradiction. Therefore, σ_2 must be uncontrollable, i.e., $\sigma_2 \in \Sigma_u$. By similar reasoning we can see that $\sigma_i \in \Sigma_u\ (\forall i \in \textbf{k-1})$. Since $\sigma_k \in \Sigma_c$ implies that the state t is not removed by the synthesis algorithm hence $\{p, q, \cdots, t\} \in G^j$ which is a contradiction to the assumption that $q \notin G^j$.
(β) Then $p \overset{s}{\mapsto}_{G^j} r \overset{t}{\mapsto}_{G^j} q$ implies $q \notin L^m \times D^m$, $s \in \Sigma_\ell^*$, and $t \in (\Sigma - \Sigma_\ell)^*$. Since $\bar{P}$ is E-observer then $p \overset{t}{\mapsto}_{\tilde{G}^N} q$ in $\tilde{G}^N$ also implies $q \notin L^m \times D^m$ thus $p \notin \tilde{G}^N$, which contradicts the initial assumption.
$(\supseteq)$ **BASE:** Let $j = 0$. By definition $G^N \subseteq L \times D = \tilde{G}^0$.

INDUCTION: Assuming that the property holds for j. It needs to be shown that it also holds for $j+1$. Let $p = \langle \ell, \eta \rangle \in G^N$. By the inductive assumption it holds that $p \in \tilde{G}^j$. Assume that $p \notin G^{j+1}$. This implies that either p is (α) uncontrollable or (β) blocking state and therefore removed by the synthesis algorithm.

(α) Then there exists $v \in \Sigma_u \cap (\Sigma - \Sigma_\ell)$ such that $p \overset{v}{\mapsto}_{\tilde{G}^j} q \notin \tilde{G}^j$ for some $q = \langle \acute{\ell}, \acute{\eta} \rangle \in L \times D$. Let a sequence of consecutive transitions in G^N be the form $p \overset{\sigma_1}{\mapsto}_{G^N} r \overset{\sigma_2}{\mapsto}_{G^N} \cdots \overset{\sigma_k}{\mapsto}_{G^N} q$ $(k \geq 1)$ such that $\sigma_i \in \Sigma_\ell$ $(i \in \textbf{k-1})$ and $\sigma_k = v$. Then immediately we can see that by definition of OCC, $\sigma_i \in \Sigma_u$. Hence, for $p \overset{\sigma_1}{\mapsto}_{G^N} r \in G^N$ we have $\sigma_1 \in \Sigma_u$. This implies that $p \notin G^N$ which is a contradiction.

(β) Then there exists $p \overset{t}{\mapsto}_{\tilde{G}^j} q$ such that $q \notin L^m \times D^m$ and $t \in (\Sigma - \Sigma_\ell)^*$. Let the corresponding sequence of consecutive transitions in G^N be the form $p \overset{s}{\mapsto}_{G^N} r \overset{t}{\mapsto}_{G^N} q$ where $s \in \Sigma_\ell^*$. Since $\bar{P}$ is E-observer and $q \notin L^m \times D^m$ implies that $p \notin G^N$, which is a contradiction. ∎

We can extend Theorem 2 to accommodate systems composed of two components.

Proposition 4. *Let E_1 and E_2 be two nonblocking EFAs along with local and uncontrollable event sets $\Sigma_\ell, \Sigma_u \subseteq \Sigma := \Sigma_1 \cup \Sigma_2$, respectively, and let $E := E_1 \| E_2$. Define the transition projections $\bar{P} : T_E \times \Sigma_\ell \to T_E$, $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1, 2)$ and let EFA $\tilde{E} = \hat{P}(E, \Sigma_\ell)$. Suppose the set of forbidden locations is $\tilde{L}_f \subset \tilde{L}$. If for $i = 1, 2$, $\bar{Q}_i$ is an E_i-observer and OCC for E_i then $supS(E, \tilde{L}_f) = supS(\tilde{E}, \tilde{L}_f) \| E$.*

Proof: See [28]. ∎

By an argument similar to that for Theorem 2 we can further extend Proposition 4 for n number of EFAs as follows.

Corollary 1. *Let E be the plant consisting of $n \geq 2$ nonblocking components. Assume the set of local and uncontrollable events $\Sigma_\ell, \Sigma_u \subseteq \Sigma := \bigcup_{i=1}^n \Sigma_i$, respectively. Define the transition projections $\bar{P} : T \times \Sigma_\ell \to T$, $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i \in \boldsymbol{n})$. Let $\tilde{E} := \hat{P}[E, \Sigma_\ell]$ be the projected plant and let $\tilde{L}_f \subset \tilde{L}$ be the set of forbidden locations. If for $i \in \boldsymbol{n}, Q_i$ is an E_i-observer and OCC for E_i, then $supS(E, \tilde{L}_f) = supS(\tilde{E}, \tilde{L}_f) \| E$.*

Proof: The proof is similar to that of Theorem 2 and Proposition 4 by considering E_2 as $E_2 \| \cdots \| E_n$. ∎

This property was pointed out by [14], [26] and later in more general form by [23], and Corollary 1 extends it to systems modeled by EFAs.

VI. Example - Manufacturing Wokrcell

Consider a manufacturing workcell borrowed from [10], consisting of three machines M1, M2, and M3, working on parts stored in two buffers B1 and B2 of size 16 and 8, respectively. To increase the practical usage and complexity of the workcell, two inspection unites TU1 and TU2 are added to randomly inspect parts from B1 and B2. Parts that are qualified will be returned to the buffers, otherwise they will be eliminated. Fig. 2 shows the workcell product flow

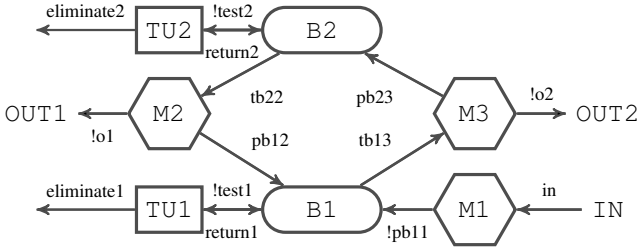


Fig. 2: The manufacturing workcell control flow.

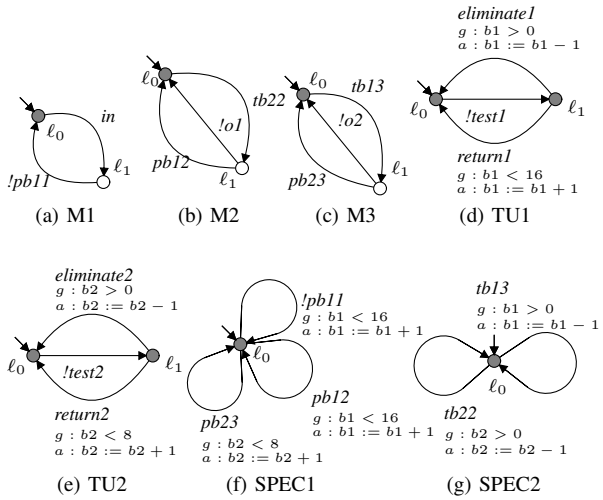


Fig. 3: EFA components of the example

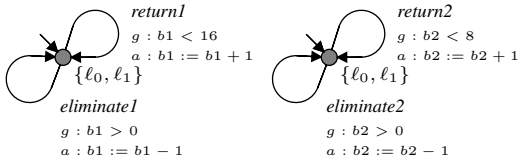


Fig. 4: Abstracted EFA models of TU1 and TU2.

and Fig. 3 illustrates the EFA components of the system in which the events with exclamation mark are the uncontrollable events and the shaded circles are the marked locations.

The domain of the variables b_1 and b_2 are, respectively, $D_1 = \{0, 1, 2, \ldots, 16\}$ and $D_2 = \{0, 1, 2, \ldots, 8\}$ that indicate the number of parts in the two buffers and their maximum capacity. B1 and B2 initially contain no part, i.e., $D^0 = \{(0, 0)\}$, and all values are marked $D^m = D_1 \times D_2$. The workcell specifications are as follows. SPEC1: buffers B1 and B2 must not overflow, i.e., a machine must not try to put a part in a buffer when it is full, formally, when $b_1 = 16$ or $b_2 = 8$. SPEC2: buffers B1 and B2 must not underflow, i.e., a machine must not try to take a part from a buffer when it is empty, formally, when $b_1 = 0$ or $b_2 = 0$. The corresponding EFAs for SPEC1 and SPEC2 are depicted in Fig. 3(f) and Fig. 3(g), respectively.

To apply the model abstraction using transition projection as mentioned earlier, first we find the local events in the system by checking the conditions in Definition 4 for all the events. The first candidates for the set of local events

TABLE I: Optimal nonblocking supervisory synthesis results of the manufacturing workcell example

	Reachable States	Supervisor States
Original Models	4896	4752
Abstracted Models	1224	1188

are $\Sigma_\ell = \{$*in, !test1, !test2, !o1, !o2*$\}$. In this set, the events "*!o1*" and "*!o2*" do not fulfill the observer property and therefore, are eliminated from the list. Also, the event "*in*" is found to be inconsistent with OCC conditions; thus it is removed. Finally, the remaining local events $\Sigma_\ell = \{$*!test1, !test2*$\}$ are used to project the EFA components. The projection of TU1 and TU2 are depicted in Fig. 4(a) and Fig. 4(b), respectively, while the rest are the same as the original ones. The optimal nonblocking and controllable guards added to the abstracted system are the same as the original system in [10]. Table I shows the result of the optimal nonblocking supervisory synthesis for both the original and the abstracted systems.

VII. Conclusion

In this paper we have extended previous work on model abstraction by natural projection with a modified observer property to include the EFA modeling formalism. Transition projection is introduced as an extension of natural projection for EFAs by directly projecting the conditional transitions. We independently compute the projection of the low-level components without regard to their mutual conflict. Subsequently, to reduce computational complexity, we compute the high-level coordinators based only on abstracted models of the low-level components. Effective and consistent model abstractions are accomplished through transition projections with the observer and OCC properties. A manufacturing workcell example demonstrates the computational effectiveness and practical usage of the proposed approach. A special case of this abstraction, including additional structural reduction, has been applied on a large-scale manufacturing workcell [30], where more than 98% of the computational time and space has been saved.

References

[1] P. Ramadge and W. Wonham, "The control of discrete event systems," *Proceedings of IEEE, Special Issue on Discrete Event Dynamic Systems*, vol. 77, no. 1, pp. 81–98, 1989.

[2] M. Fabian and A. Hellgren, "PLC-based Implementation of Supervisory Control for Discrete Event Systems," in *37th Decision and Control*, Tampa, FL, USA, 1998.

[3] X.-R. Cao, G. Cohen, A. Giua, W. Wonham, and J. H. van Schuppen, "Unity in Diversity, Diversity in Unity: Retrospective and Prospective Views on Control of Discrete Event Systems," *Discrete Event Dynamic Systems*, vol. 12, pp. 253–264, 2002.

[4] V. Rusu, H. Marchand, and T. Jéron, "Automatic Verification and Conformance Testing for Validating Safety Properties of Reactive Systems," *FM 2005: Formal Methods*, pp. 189–204, 2005.

[5] M. Skoldstam, K. Åkesson, and M. Fabian, "Modeling of discrete event systems using finite automata with variables," *2007 46th IEEE Conference on Decision and Control*, pp. 3387–3392, 2007.

[6] B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman, and K. Akesson, "Sequence Planning for Integrated Product, Process and Automation Design," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 791–802, Oct. 2010.

[7] M. R. Shoaei, B. Lennartson, and S. Miremadi, "Automatic generation of controllers for collision-free flexible manufacturing systems," in *IEEE International Conference on Automation Science and Engineering*, Aug. 2010, pp. 368–373.

[8] A. Vahidi, M. Fabian, and B. Lennartson, "Efficient supervisory synthesis of large systems," *Control Engineering Practice*, vol. 14, no. 10, pp. 1157–1167, Oct. 2006.

[9] S. Miremadi, B. Lennartson, and K. Åkesson, "BDD-based Supervisory Control on Extended Finite Automata," in *Proceedings of the 7th IEEE Conference on Automation Science and Engineering*, 2011.

[10] L. Ouedraogo, R. Kumar, R. Malik, and K. Åkesson, "Nonblocking and Safe Control of Discrete-Event Systems Modeled as Extended Finite Automata," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 560–569, Jul. 2011.

[11] K. Åkesson, M. Fabian, H. Flordal, and R. Malik, "Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems," in *Proceedings of WODES'08*, Ann Arbor, MI, USA, 2006, pp. 384–385.

[12] P. Gohari and W. Wonham, "On the complexity of supervisory control design in the RW framework." *IEEE transactions on systems, man, and cybernetics.*, vol. 30, no. 5, pp. 643–52, Jan. 2000.

[13] K. Wong and W. Wonham, "Modular control and coordination of discrete-event systems," *Discrete Event Dynamic Systems*, vol. 8, no. 3, pp. 247–297, Oct. 1998.

[14] M. H. Queiroz, J. E. R. Cury, and M. de Queiroz, "Modular control of composed systems," in *American Control Conference*, vol. 6, no. June. American Autom. Control Council, Jun. 2000, pp. 4051–4055.

[15] L. S-H. and K. Wong, "Structural decentralized control of concurrent discrete-event systems," *European Journal of Control*, pp. 1125–1134, 2002.

[16] K. Schmidt, J. Reger, and T. Moor, "Hierarchical control for structural decentralized DES," in *Discrete event systems 2004 (WODES'04)*, Elsevier Science. Elsevier Science, 2004, p. 279.

[17] K. Wong and W. Wonham, "Hierarchical Control of Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 6, no. 3, pp. 241–273, Jul. 1996.

[18] R. J. Leduc, M. Lawford, W. Wonham, and B. A. Brandin, "Hierarchical interface-based supervisory Control-part I: serial case," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1322–1335, Sep. 2005.

[19] R. Leduc, M. Lawford, and W. Wonham, "Hierarchical interface-based supervisory control-part II: parallel case," in *IEEE Transactions on Automatic Control*, vol. 50, no. 9, Sep. 2005, pp. 1336–1348.

[20] S. Mohajerani, R. Malik, and M. Fabian, "Nondeterminism avoidance in compositional synthesis of discrete event systems," in *2011 IEEE International Conference on Automation Science and Engineering*. IEEE, Aug. 2011, pp. 19–24.

[21] K. Wong and W. Wonham, "On the Computation of Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 14, no. 1, pp. 55–107, Jan. 2004.

[22] L. Feng and W. Wonham, "Supervisory Control Architecture for Discrete-Event Systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1449–1461, Jul. 2008.

[23] ——, "On the Computation of Natural Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 20, no. 1, pp. 63–102, Oct. 2008.

[24] F. Lin and W. Wonham, "Decentralized Supervisory Control of Discrete Event Systems," *Information Sciences*, vol. 44, pp. 199–224, 1988.

[25] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer, 2008.

[26] W. Wonham, *Supervisory Control of Discrete Event Systems*, Toronto, Canada, 2012.

[27] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., ser. Series in Computer Science. Addison-Wesley, 2001.

[28] M. R. Shoaei, L. Feng, and B. Lennartson, "Supervisory Control of Extended Finite Automata using Transition Projection," Chalmers University of Technology, Tech. Rep., 2012. [Online]. Available: http://publications.lib.chalmers.se/cpl/record/index.xsql?pubid=155706

[29] K. Wong, "On the Complexity of Projections of Discrete-Event Systems," in *Proc. the 4th international workshop on discrete event systems*, 1998, pp. 201–206.

[30] M. R. Shoaei, S. Miremadi, K. Bengtsson, and B. Lennartson, "Reduced-order synthesis of operation sequences," in *ETFA2011*. IEEE, Sep. 2011, pp. 1–8.