This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Exploiting Distinguishers in Local Modular Control of Discrete-Event Systems

Marcelo Teixeira, José E. R. Cury, and Max H. de Queiroz

***Abstract*— Local modular control (LMC) is an approach to the supervisory control theory (SCT) of discrete-event systems that exploits the modularity of plant and specifications. Recently, *distinguishers* and *approximations* have been associated with SCT to simplify modeling and reduce synthesis effort. This paper shows how advantages from LMC, distinguishers, and approximations can be combined. Sufficient conditions are presented to guarantee that local supervisors computed by our approach lead to the same global closed-loop behavior as the solution obtained with the original LMC, in which the modeling is entirely handled without distinguishers. A further contribution presents a modular way to design distinguishers and a straightforward way to construct approximations to be used in local synthesis. An example of manufacturing system illustrates our approach.**

***Note to Practitioners*—*Distinguishers* and *approximations* are alternatives to simplify modeling and reduce synthesis cost in SCT, grounded on the idea of event-refinements. However, this approach may entangle the modular structure of a plant, so that LMC does not keep the same efficiency. This paper shows how *distinguishers* and *approximations* can be locally combined such that synthesis cost is reduced and LMC advantages are preserved.**

***Index Terms*— Approximations, discrete-event systems (DESs), distinguishers, local modular control (LMC).**

I. Introduction

Supervisory control theory (SCT) [1] formally describes the synthesis of supervisors (controllers) for *discrete-event systems* (DESs), mathematically grounded on *finite-state automata* formalism [2]. Two major concerns in SCT are the computational cost imposed by the algorithms that synthesize supervisors and the difficulty faced when expressing the rules to be fulfilled by the system under control [3], [4].

The first problem arises when the number of states of a DES model exponentially grows as a function of the number of components in the system. The second problem is faced by control engineers when they cannot easily express a certain control requirement as an automaton with a reasonable size.

Synthesis and modeling issues have been addressed in SCT using alternatives, such as *hierarchical control* [5]–[7], *symbolic computation* [8]–[10], *abstractions* [3], [11], [12], and so on. Yet, these approaches in general do not combine both modeling and synthesis advantages into the same framework.

Local modular control (LMC) [13], for example, structures the synthesis of supervisors in a localized manner, which tends to reduce synthesis effort. In LMC, each specification defines a local problem, which is then solved by using a local plant, composed by an appropriate subset of plant models.

Although LMC allows reducing the computational cost of synthesis, this does not generally simplify modeling tasks. In this sense, the use of *distinguishers* [4], [14] can be considered. This approach consists of refining the alphabet of events of a DES model into a new set, in a way to identify instances of the original events in the system. If properly chosen, this larger alphabet can simplify the modeling of control specifications, at the price of enriching the plant with the distinguisher model. When properly addressed, the use of approximations with distinguishers can also reduce the synthesis complexity.

This paper subsumes previous results [4], [15], [16] to show the combined use of distinguishers, approximations, and LMC. It is shown that distinguishers are helpful to solve some local control problems, but their use may complexify the local plant model of other specifications. Thus, we provide conditions under which supervisors can be locally synthesized in a combined way, either using the distinguishers or not, whenever appropriate. Maps are provided to merge local supervisors into the same alphabet for implementation.

In spite of the modeling benefits, for the combined approach to achieve the same global controlled behavior as in the original LMC (without distinguishers), it also requires equivalent computational effort when solving the original problem. That is, the combined synthesis is useful to simplify the modeling tasks but it is not directly advantageous computationally. In this sense, we also show that the combined synthesis can be extended to the use of outer approximations of the distinguisher model. An example of a small manufacturing system control illustrates the approach.

This document is structured as follows. Section II recalls concepts from SCT, LMC, distinguishers, and approximations, which is illustrated in Section III. Section IV generalizes and rewrites results in [15] and [16], which are combined in Section V. Conclusions are discussed in Section VI.

II. Preliminaries

SCT describes the supervisor synthesis for DESs [1]. In this approach, a DES plant is modeled by a finite-state automaton $G = (\Sigma, X, x_0, g, X_m)$, such that $\Sigma = \Sigma_c \dot\cup \Sigma_u$ is the alphabet resulting from the union of disjoint sets of controllable (Σ_c) and uncontrollable (Σ_u) events, X is the set of states, $x_0 \in X$

Manuscript received December 12, 2016; revised May 28, 2017; accepted January 11, 2018. This paper was recommended for publication by Associate Editor M. Fabian and Editor S. Reveliotis upon evaluation of the reviewers' comments. This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq). *(Corresponding author: Marcelo Teixeira.)*

M. Teixeira is with the Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Pato Branco 85505031, Brazil (e-mail: marceloteixeira@utfpr.edu.br).

J. E. R. Cury and M. H. de Queiroz are with the Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis 88040-900, Brazil (e-mail: jose.cury@ufsc.br; max.queiroz@ufsc.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2018.2793963

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

is the initial state, $g : X \times \Sigma \to X$ is a partial transition function, and $X_m \subseteq X$ is the subset of marked states.

The language generated by a plant G is denoted by $L(G) \subseteq \Sigma^*$ and represents the set of all strings of events possible to occur in G, while $L_m(G) \subseteq L(G)$ denotes the language marked by G, containing the strings that represent the completed tasks. The parallel composition [2] of two automata, G^a and G^b, is denoted by $G^a \| G^b$, and the same notation is used for the parallel composition of languages. For any automaton G, the set of events used to model G is denoted by Σ^G. For any given alphabet Σ, H_Σ denotes the automaton with a single (marked) state, such that $L_m(H_\Sigma) = \Sigma^*$.

A *(marker) supervisor* S is a map $S : L(G) \to 2^\Sigma$ together with a language $L_S \subseteq L_m(G)$ such that, for each string s generated by G, the control action of S over G enables events of $S(s) \subseteq \Sigma$, with $\Sigma_u \subseteq S(s)$, and marks strings $s \in L_S$. We assume that S/G is an automaton representing the controlled system (*closed-loop behavior*), containing the strings of $L(G)$ that survive to the supervision of S, with $L_m(S/G) = L(S/G) \cap L_S$. S is said to be *nonblocking* when $L(S/G) = \overline{L_m(S/G)}$, i.e., when all strings surviving under control are prefixes of complete tasks in the closed-loop system. The *supervisory control problem* (SC-P) [17] can then be presented as follows.

Problem 1 [SC-P]: Given a plant G defined on Σ and a specification $E \subseteq \Sigma^*$ for G, defining a desired behavior $K = E \cap L_m(G)$, find a nonblocking supervisor S, such that $L_m(S/G) \subseteq K$.

A language $K \subseteq \Sigma^*$ is said to be *controllable* with respect to G, when $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$. Controllability of a nonempty language $K \subseteq L_m(G)$ is the necessary and sufficient condition for the existence of a nonblocking supervisor S, which satisfies a specification $K \subseteq L_m(G)$, i.e., $L_m(S/G) = K$ [17]. In this case, S such that $L_m(S/G) = K$ can be implemented by any automaton, and V such that $K = L_m(V) \cap L_m(G)$ and $\overline{K} = L(V) \cap L(G)$. The control action of S disables eligible events in $L(G)$ that are not eligible in $L(V)$, after a string $s \in L(S/G)$. Marking actions correspond to marking strings that are marked in both V and G.

Let $\mathcal{C}(K, G)$ be the set of languages in K, which is controllable with respect to G. $\mathcal{C}(K, G)$ has a unique supremal element, denoted by $\sup\mathcal{C}(K, G)$, that represents the least restrictive behavior possible to be implemented by a nonblocking supervisor S controlling G. In this case S, such that $L_m(S/G) = \sup\mathcal{C}(K, G)$ and $L(S/G) = \overline{\sup\mathcal{C}(K, G)}$, is an optimal solution to the SC-P.

A. Local Modular Control of DESs

In monolithic synthesis, the *state-space explosion* problem may become a barrier for the SCT to be adopted in the industry. Modularization is an alternative to exploit the subsets of components in synthesis, which reduces computational effort and enlarges the range of applications that can be handled by SCT.

Hierarchical interface-based [18] and compositional synthesis [12] are the examples of approaches that exploit the modularity of DESs. In this paper, the LMC [19] is exploited. In the LMC, the plant is assumed to be modeled as a set of subsystems that must be coordinated according to a set of local specifications. Then, the synthesis of local supervisors is processed using only subsystems directly affected by events from its respective specification. Synchronism of automata is the key to formally define the LMC problem.

Definition 1: A set of automata $A = \{A^i\},\ i = 1, \cdots, n$ is said to be *asynchronous*, if for all $A^j, A^k \in A$, such that $j \neq k$, $\Sigma^{A^j} \cap \Sigma^{A^k} = \emptyset$. Otherwise, A is said to be *synchronous*.

A *product system* (PS) [1] is a model for a DES represented by an asynchronous set of automata. A PS can still be obtained from a synchronous set of automata by composing the models that share events, such that a PS version exists for any set of automata. Given a set of automata A, the optimal PS (OPS) of A can be obtained as the set of automata $A' = \text{async}(A)$, where each automaton of A' is the composition of a maximal subset of automata of A pairwise sharing events [19].

Now, consider that a DES plant is modeled by a set $\mathcal{GG}$ of automata $G^{j'}$, for $j' \in J' = \{1, \cdots, n'\}$, with events in Σ. Define $E^i \subset \Sigma^{E^i*}$, for $i \in I = \{1, \cdots, m\}$, as specifications to the plant modeled with events in $\Sigma^{E^i} \subset \Sigma$. The global plant would then be given by $G = \|_{j' \in J'} G^{j'}$ and monolithic synthesis would be conducted on $K = (\|_{i \in I} E^i) \| L_m(G)$.

Differently, LMC synthesis is conducted locally, using only a specific part of G. For the set $\mathcal{GG}$ of plants, let $\mathcal{G} = \text{async}(\mathcal{GG})$ be the OPS of $\mathcal{GG}$. $\mathcal{G}$ comprises asynchronous automata G^j, $j \in J = \{1, \cdots, n\}$. Then, plants to be associated with E^i for local synthesis are composed in $G^i_{loc} = \|_{j \in J_i} G^j$, where $J_i = \{j \in J \mid \Sigma^{G^j} \cap \Sigma^{E^i} \neq \emptyset\}$. That is, G^i_{loc} composes only subsystems affected by events from E^i and local specifications are given by $K^i_{loc} = E^i \| L_m(G^i_{loc})$.

Definition 2: For a specification E^i and an OPS $\mathcal{G}$ of a set of DES plants $\mathcal{GG}$, define a local plant as $\text{LP}(\mathcal{G}, E^i) = G^i_{loc}$.

Remark that, although $\mathcal{G}$ is asynchronous, this may not be the case for the resulting set of G^i_{loc} plants. Let S^i_{loc}, for $i \in I$, be local supervisors such that $S^i_{loc} : L(G^i_{loc}) \to 2^{\Sigma^{G^i_{loc}}}$ and $L_{S^i_{loc}} \subseteq L_m(G^i_{loc})$. When acting over the global plant, each S^i_{loc} ignores and permanently enables the events in $\Sigma - \Sigma^{G^i_{loc}}$ by default. So, the concurrent action of S^i_{loc} over G is such that $\wedge_{i \in I} S^i_{loc}/G = \|_{i \in I} S^i_{loc}/G^i_{loc}$. The LMC problem can then be stated as follows.

Problem 2 (LMC-P [19]): Let $\mathcal{G}$ be an OPS of a DES plant, G be the global plant, and $E = \|_{i \in I} E^i$ be a specification for G, such that K is modeled as in SC-P. Find a set of local supervisors S^i_{loc}, for G^i_{loc}, such that $\wedge_{i \in I} S^i_{loc}$ is nonblocking with respect to G and $L_m(\wedge_{i \in I} S^i_{loc}/G) \subseteq K$.

For each specification E^i, $i \in I$, the synthesis of a local supervisor S^i_{loc} can proceed as a local instance of the SC-P, such that $L_m(S^i_{loc}/G^i_{loc}) = \sup\mathcal{C}(K^i_{loc}, G^i_{loc})$. However, it can be shown that the conjunction of supervisors can be blocking, even when they are individually nonblocking [20]. In this case, the supervisors are said to be conflicting. The necessary and sufficient condition for nonconflict of local modular supervisors is $\|_{i \in I} \overline{L_m(S^i_{loc}/G^i_{loc})} = \overline{\|_{i \in I} L_m(S^i_{loc}/G^i_{loc})}$.

In words, every string in the synchronous composition of the local supervision is a prefix of a marked string enabled by all supervisors in their respective local plants. In addition

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to nonblocking, it can be shown [19] that nonconflict ensures global optimality, i.e., $\sup\mathcal{C}(K, G) = \|_{i\in I} \sup\mathcal{C}(K^i_{loc}, G^i_{loc})$.

Therefore, if the set of optimal local modular supervisors is nonconflicting, it is a minimally restrictive solution to LMC-P. If it is not, an additional supervisor (coordinator) can be used to solve conflict [21].

B. SCT With Distinguishers

For a DES modeled by an automaton G with events in Σ, assume that each event $\sigma \in \Sigma$ is a mask for events of a set $\Delta^\sigma \neq \emptyset$, chosen in a manner to identify different instances of the occurrence of σ in the system. Then, Σ is a set of masks for events in $\Delta = \Delta_c \cup \Delta_u$, where $\Delta_u = \bigcup_{\sigma\in\Sigma_u} \Delta^\sigma$ and $\Delta_c = \bigcup_{\sigma\in\Sigma_c} \Delta^\sigma$.

The relationship between alphabets Σ and Δ is defined by the following maps. Let $\Pi : \Delta^* \to \Sigma^*$ be a *masking map*, such that $\Pi(\epsilon) = \epsilon$ and $\Pi(t\delta) = \Pi(t)\sigma$, for $t \in \Delta^*$, $\delta \in \Delta^\sigma$, and $\sigma \in \Sigma$. For any language $L \subseteq \Delta^*$, Π can be extended as $\Pi(L) = \{s \in \Sigma^* \mid \exists t \in L, \Pi(t) = s\}$. Also, for any string $s \in \Sigma^*$, one can define the *inverse masking map* Π^{-1}: $\Sigma^* \to 2^{\Delta^*}$ by $\Pi^{-1}(s) = \{t \in \Delta^* \mid \Pi(t) = s\}$. Remark that an automaton $\Pi^{-1}(G)$, generating the language $\Pi^{-1}[L(G)]$, can be obtained by simply replacing the events of each transition in G by the corresponding sets of refined events. Now, a distinguisher can be defined as follows.

Definition 3: Given a masking map $\Pi : \Delta^* \to \Sigma^*$, let $L_d = \overline{L_d} \subseteq \Delta^*$, with $\Pi(L_d) = \Sigma^*$, be the language with all possible distinctions of strings in Σ^*. A distinguisher $D : \Sigma^* \to 2^{L_d}$ is defined by $D(s) = \Pi^{-1}(s) \cap L_d$.

That is, a distinguisher D maps a string of masks into the corresponding strings of refined events from L_d. We assume in the following that D belongs to a particular class of distinguishers, so-called *predictable* distinguishers, case in which each string of Σ^* is mapped into exactly one string in Δ^*, i.e., $|D(s)| = 1$, for all $s \in \Sigma^*$. Modeling the language L_d is a crucial step of the approach with distinguishers, which is properly illustrated in Section III.

Distinguishers can be extended from strings to languages as usual. For a DES modeled by an automaton G, we denote by G_d [or $D(G)$] the automaton such that $L(G_d) = D[L(G)]$ and $L_m(G_d) = D[L_m(G)]$ and by E_d a specification modeled in Δ. Now, for a supervisor S_d defined by a map $S_d : L(G_d) \to 2^\Delta$ together with a language $L_{S_d} \subseteq L_m(G_d)$, we define the supervisor $\Pi(S_d) : L(G) \to 2^\Sigma$ and the language $L_S \subseteq L_m(G)$ by $\Pi(S_d)(s) = \{\Pi(\delta) \in \Sigma \mid \delta \in S_d(t), \{t\} = D(s)\}$, and $L_S = \Pi(L_{S_d})$.

From the above-mentioned definition, $\Pi(S_d)$ can be seen as the undistinguished version of the supervisor S_d. After a string $s \in L(G)$, $\Pi(S_d)$ will enable the set of masks of events in $S_d(t)$, with t being the distinguished version of s. Also, the string s is marked by $\Pi(S_d)$, if t is marked by S_d.

Now, the SC-P can be modified to consider the introduction of a distinguisher (namely DSC-P).

Problem 3 (DSC-P [4]): Given a plant G with alphabet Σ, a predictable distinguisher $D : \Sigma^* \to 2^{\Delta^*}$, and a specification $E_d \subseteq \Delta^*$, defining a desired behavior $K = \Pi[E_d \cap L_m(G_d)]$ for G, find a nonblocking supervisor S_d for G_d, such that $\Pi(S_d)$ is a nonblocking supervisor for G, with $L_m[\Pi(S_d)/G] \subseteq K$.

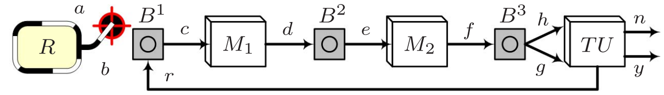

Fig. 1. Manufacturing system with material feedback.

The idea in the DSC-P is to distinguish G by D, in a way that $E \subseteq \Sigma^*$ could be more easily written as E_d with events in Δ. In this way, $K_d = E_d \cap L_m(G_d)$ is an alternative representation of the desired behavior $K = E \cap L_m(G)$, with $\Pi(K_d) = K$. Then, the DSC-P consists of finding a supervisor S_d from K_d and G_d, such that its undistinguished version $\Pi(S_d)$ is a nonblocking supervisor for G that satisfies K. It can be shown [4] that, if $L_m(S_d/G_d) = \sup\mathcal{C}(K_d, G_d)$ then $L_m(\Pi(S_d)/G) = \sup\mathcal{C}(K, G)$.

C. Synthesis With D-Approximations

In the following, we describe how computational savings can be provided in synthesis with distinguishers.

Definition 4 [4]: For a plant G and a predictable distinguisher $D : \Sigma^* \to 2^{L_d}$, if $D_a : \Sigma^* \to 2^{L_{da}}$ is such that $L_d \subseteq L_{da} \subseteq \Delta^*$, then $G_a = D_a(G)$ is a *D-approximation*, for $G_d = D(G)$.

By choosing L_{da} as a proper outer approximation of L_d, we may expect that G_a has fewer states than G_d. For instance, the coarsest D-approximation G_a, such that $L_{da} = \Delta^*$, has the same number of states as the unrefined plant G. Optimality in synthesis with *D-approximations* can be verified as follows.

Definition 5 [4]: Given $L_1 \subset L_2 \subseteq \Delta^*$, a masking map $\Pi : \Delta^* \to \Sigma^*$, and a set of masks $\Lambda \subseteq \Sigma$, L_1 is Λ-*preserving* with respect to L_2, if $\forall\sigma \in \Lambda$ and $t \in \overline{L_1}$, $t\Delta^\sigma \cap \overline{L_2} \neq \emptyset \to t\Delta^\sigma \cap \overline{L_1} \neq \emptyset$.

That is, L_1 is Λ-*preserving* with respect to L_2 if, for every mask $\sigma \in \Lambda$, whenever a refined event in Δ^σ is eligible in $\overline{L_2}$, at least one instance of Δ^σ is also eligible in $\overline{L_1}$. For languages L_1 and L_2 in Δ^* and $\Lambda \subseteq \Sigma$, define the set $\mathcal{P}_\Lambda(L_1, L_2) = \{K \subseteq L_1 \mid K \text{ is } \Lambda\text{-}Preserving \text{ with respect to } L_2\}$. It can be shown that $\mathcal{P}_\Lambda(L_1, L_2)$ has a supremal element, denoted $\sup\mathcal{P}_\Lambda(L_1, L_2)$, which can be computed in polynomial time in the number of states of the automata models for L_1 and L_2 [14].

Proposition 1 [4]: Given G_d and E_d as in DSC-P, let G_a be a D-approximation for G_d. For $K_a = E_d \cap L_m(G_a)$, if $\sup\mathcal{C}(K_a, G_a)$ and L_d are nonconflicting, a nonblocking supervisor S_a, such that $L_m(S_a/G_d) = \sup\mathcal{C}(K_a, G_a) \cap L_d$ is a solution for DSC-P. Moreover, $\sup\mathcal{C}(K_a, G_a) \cap L_d \subseteq \sup\mathcal{C}(K_d, G_d) \subseteq \sup\mathcal{P}_{\Sigma_u}[K_a, L(G_a)] \cap L_d$.

Proposition 1 shows that a supervisor S_a can be a solution to DSC-P, although it may be suboptimal. The proximity between $\sup\mathcal{C}(K_a, G_a)$ and $\sup\mathcal{P}_{\Sigma_u}[K_a, L(G_a)]$ gives a measure of how far $\sup\mathcal{C}(K_a, G_a) \cap L_d$ can be from the optimal solution $\sup\mathcal{C}(K_d, G_d)$. In particular, if they are equal, then $\sup\mathcal{C}(K_d, G_d) = \sup\mathcal{C}(K_a, G_a) \cap L_d$.

III. Example

Fig. 1 shows an example of manufacturing system that illustrates our approach. Examples in [3], [4], [15], [16], and [22]–[24] are also applicable.

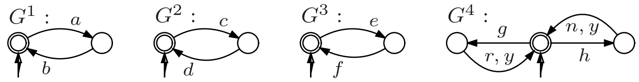

Fig. 2. Models for the subsystems R, M_1, M_2, and TU.

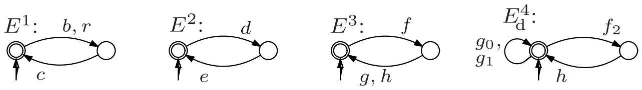

Fig. 3. Specification models E^1, E^2, and E^3, in Σ, and E_{d}^4, in Δ.

The system is composed by a robot (R), two machines (M_1 and M_2), and a test unit (TU), connected by intermediate unitary buffers. The robot R takes workpieces from the storage (event a) and inserts them on the buffer B_1 (event b). Then, machines M_1 and M_2 pickup workpieces from the input buffer, manufacture, and release them to the respective output buffer. Finally, TU tests the quality of the workpieces, deciding by their approval (event y), rejection (event n) or rework in M_1 (event r). The plant can be modeled by the automata G^1, G^2, G^3, and G^4, in Fig. 2.

Events a, c, and e model an operation starting in R, M_1, and M_2, respectively, while events b, d, and f model their conclusion. Event g models a general test in TU, which leads to approval (y) or rework (r) in M_1 (y or r), while event h models a final test, after which the workpieces are approved (y) or rejected (n). The composition $G = G^1 \| G^2 \| G^3 \| G^4$ models the uncontrolled system behavior, where it is assumed that $\Sigma_{\mathrm{c}} = \{a, c, e, g, h\}$ and $\Sigma_{\mathrm{u}} = \{b, d, f, n, y, r\}$.

The control objective is to avoid *overflow* and *underflow* in the buffers B^i, $i = 1, 2, 3$, which can be modeled by E^i, as shown in Fig. 3. We also consider an extra specification, E^4, aimed to allow a final test only when a workpiece has been rejected twice in the general test. While E^1, E^2, and E^3 require only two states each to be modeled, designing E^4 in the alphabet Σ turns out to be a very complex engineering task, as the system can concurrently process multiple workpieces.

This task can nevertheless be simplified by the use of distinguishers. Note that modeling E^4 requires memorizing the number of times that the same workpiece has been manufactured. As up to three workcycles are allowed, then we refine: $\Delta^c = \{c_0, c_1, c_2\}$; $\Delta^d = \{d_0, d_1, d_2\}$; $\Delta^e = \{e_0, e_1, e_2\}$; $\Delta^f = \{f_0, f_1, f_2\}$; $\Delta^g = \{g_0, g_1\}$; $\Delta^r = \{r_1, r_2\}$; and $\Delta^\sigma = \{\sigma\}, \forall \sigma \in \{a, b, h, n, y\}$. Semantically, for each $\sigma \in \Sigma$, σ_0 means the first manufacture (no rework), while σ_1 and σ_2 model first and second reworks, respectively. Then, the refined alphabet is given by $\Delta = \dot{\bigcup}_{\sigma \in \Sigma} \Delta^\sigma$.

In Δ, the specification E^4 can be expressed by the 2-state automaton E_{d}^4 in Fig. 3, which prohibits the final test (event h) until the third buffer has a workpiece rejected twice (event f_2), and prohibits a general test (events g_0 and g_1) otherwise.

Although E^4 is now expressed by a simple automaton, the number of events associated with each original event in the system is expanded, so that their occurrences have to be *distinguished*. This can be achieved by associating the modular automata in Fig. 4 to the plant.

Each automaton distinguishes a single instance of refinement from the others. For example, $H_{\mathrm{d}c}^1$ distinguishes c_1 (from c_0 and c_2) after an event r_1 (first rework); $H_{\mathrm{d}c}^2$ distinguishes c_2 (from c_0 and c_1) after an event r_2 (second rework); and c_0 is initially enabled in both $H_{\mathrm{d}c}^1$ and $H_{\mathrm{d}c}^2$ (first cycle) and disabled afterward. Then, d_0, d_1, and d_2 are distinguished by $H_{\mathrm{d}d}^1$ and $H_{\mathrm{d}d}^2$, based on the precedence of c_0, c_1, c_2, and so on.

In this case, $H_{\mathrm{d}} = H_{\mathrm{d}c}^1 \| \cdots \| H_\Delta$, for H_Δ such that $L_m(H_\Delta) = \Delta^*$, models a predictable distinguisher with $L(H_{\mathrm{d}}) = L_{\mathrm{d}}$. From H_{d}, an equivalent version of the specification E_{d}^4 could be constructed in Σ as $E^4 = \Pi(E_{\mathrm{d}}^4 \| H_{\mathrm{d}})$, which in this case corresponds to 1.310-state automaton.

TABLE I

Number of States—SC-P and DSC-P

G	E	K	V
24	1.706	7.220	212
G_{d}	E_{d}	K_{d}	V_{d}
2.160	12	7.220	212
G_{ca}	E_{d}	K_{ca}	V_{ca}
24	12	288	62
G_{a}	E_{d}	K_{a}	V_{a}
36	12	432	81

A. Monolithic Synthesis

Table I (rows 1 and 2) presents synthesis results for SC-P and DSC-P, where both versions of the problem lead to equivalent results (V and V_{d} with 212 states), which are also obtained with the same computational cost (K and K_{d} with 7.220 states).

Despite the modeling benefits brought by distinguishers, synthesis cost remains the same as in SC-P. Approximations are approached next to simplify synthesis. We first use the coarsest approximation G_{ca}, such that $L(G_{\mathrm{ca}}) = D_{\mathrm{ca}}^{L(G)} = \Pi^{-1}[L(G)]$, i.e., distinctions were entirely removed from G_{d} and L_{dca} has been modeled by an automaton $H_{\mathrm{dca}} = H_\Delta$. Then, the global specification is given by $K_{\mathrm{ca}} = E_{\mathrm{d}} \cap L_m(G_{\mathrm{ca}})$, and V_{ca} is an automaton for $\sup\mathcal{C}(K_{\mathrm{ca}}, G_{\mathrm{ca}})$ (third row).

To synthesize V_{ca}, an automaton with only 288 states has been used, instead of 7.220 as when obtaining V_{d}. However, since $\sup\mathcal{C}(K_{\mathrm{ca}}, G_{\mathrm{ca}}) \neq \sup\mathcal{P}_{\Sigma_{\mathrm{u}}}[K_{\mathrm{ca}}, L(G_{\mathrm{ca}})]$, we cannot conclude on the optimality of S_{ca} from Proposition 1 without computing $\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$.

In order to obtain a solution that is proven to be optimal, part of H_{d} can be included in synthesis. When H_{d} is omitted, multiple workpieces are prevented to be simultaneously processed in every cycle, in order to guarantee the specification E_{d}^4. However, when a workpiece is in its last processing cycle, new workpieces should be allowed to enter, as there is certainly no more rework to occur. Such information is particularly provided by $H_{\mathrm{d}f}^2$, which distinguishes the instances of f to E_{d}^4.

Now let G_{a}, with $L(G_{\mathrm{a}}) = D_{\mathrm{a}}^{L(G)} = \Pi^{-1}[L(G)] \cap L_{\mathrm{da}}$, be the approximation for G_{d} obtained by applying a distinguisher D_{a} over G, such that D_{a} is modeled by $H_{\mathrm{d}f}^2 \| H_\Delta$. The model for $K_{\mathrm{a}} = E_{\mathrm{d}} \cap L(G_{\mathrm{a}})$ has 432 states, and V_{a} such that $L_m(V_{\mathrm{a}}) = \sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$ has 81 states (fourth row).

By checking that $V_{\mathrm{a}} \| H_{\mathrm{d}}$ is nonblocking, we know that $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$ and L_{d} are nonconflicting. Therefore, $S_{\mathrm{a}} = \sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$ is a solution for SCP-D, by Proposition 1. Moreover, now $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) = \sup\mathcal{P}_{\Sigma_u}[K_{\mathrm{a}}, L(G_{\mathrm{a}})]$, and then

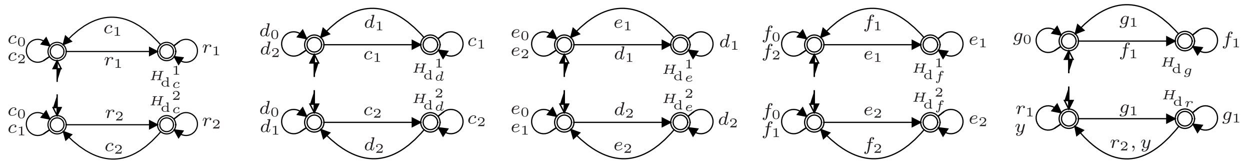

Fig. 4. Modular representation of the distinguisher $H_{\mathrm{d}} = H_{\mathrm{d}c}^{1} \| H_{\mathrm{d}c}^{2} \| H_{\mathrm{d}d}^{1} \| H_{\mathrm{d}d}^{2} \| H_{\mathrm{d}e}^{1} \| H_{\mathrm{d}e}^{2} \| H_{\mathrm{d}f}^{1} \| H_{\mathrm{d}f}^{2} \| H_{\mathrm{d}g} \| H_{\mathrm{d}r} \| H_{\Delta}$ with 788 states.

TABLE II

NUMBER OF STATES—LMC-P

$i \in I$	G_{loc}^i	E^i	K_{loc}^i	V_{loc}^i
1	12	2	24	14
2	4	2	8	6
3	6	2	12	9
4	12	**1.310**	2.662	2.400
Nonconflict test: $G_{conf} = \|_{i \in I} V_{loc}^i$				276

TABLE III

NUMBER OF STATES—LMC-P WITH DISTINGUISHERS

$i \in I$	$G_{\mathrm{d}}{}_{loc}^i$	E_{d}^i	$K_{\mathrm{d}}{}_{loc}^i$	$V_{\mathrm{d}}{}_{loc}^i$
1	3.412	2	**3.728**	1.296
2	1.706	2	**1.778**	725
3	1.706	2	**2.550**	1.174
4	1.706	2	2.662	2.400
Nonconflict test: $\|_{i \in I} V_{\mathrm{d}}{}_{loc}^i$				276

V_{a} and H_{d} can be used to implement an optimal control solution to the manufacturing system, such that $L_m[\Pi(S_{\mathrm{a}}/G_{\mathrm{d}})] = \Pi[\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}] = \sup\mathcal{C}[\Pi(K_{\mathrm{d}}), G]$.

B. Local Modular Synthesis

Let the asynchronous set of automata $\mathcal{G} = \mathcal{G}\mathcal{G} = \{G^1, G^2, G^3, G^4\}$ be as in Fig. 2 and let E^1, E^2, E^3, and E_{d}^4 be as in Fig. 3, such that $E^4 = \Pi(E_{\mathrm{d}}^4 \| H_{\mathrm{d}})$. For $I = \{1, \cdots, 4\}$, we calculate a local plant G_{loc}^i for each specification E^i. Results are presented in Table II.

Note that the most complex step in LMC-P is the computation of V_{loc}^4, with 2.662 states, while the SC-P synthesis involves 7.220 states (Table I). Besides, each supervisor V_{loc}^i is nonblocking with respect to G_{loc}^i, by construction. However, it can be checked that their joint action over G is conflicting, since the automaton $G_{\mathrm{conf}} = \|_{i \in I} V_{loc}^i$, with 276 states, is blocking. From [25], a *coordinator* V_{coord} with 10 states has been obtained from the automaton G_{conf}, such that $L_m(V_{\mathrm{coord}}/G_{\mathrm{conf}}) = \sup\mathcal{C}[L_m(G_{\mathrm{conf}}), G_{\mathrm{conf}}]$ and the set $\{V_{loc}^1, \cdots, V_{loc}^4, V_{\mathrm{coord}}\}$ is nonconflicting.

C. Naive Local Modular Synthesis With Distinguishers

Now, we show how the LMC-P can be approached in a straightforward way to integrate distinguishers. For the set $\mathcal{G} = \{G^1, G^2, G^3, G^4\}$ of automata modeling a DES plant with events in Σ, let $\mathcal{G}_{\mathrm{a}} = \{G_{\mathrm{a}}^1, G_{\mathrm{a}}^2, G_{\mathrm{a}}^3, G_{\mathrm{a}}^4\}$ be the set of corresponding automata fully mapped in Δ, such that $G_{\mathrm{a}}^j = \Pi^{-1}(G^j)$, for $j = 1, \cdots, 4$, and let H_{d} be the distinguisher model presented in Fig. 4. By $\mathcal{G}_{\mathrm{d}} = \mathrm{async}(\{\mathcal{G}_{\mathrm{a}}, H_{\mathrm{d}}\})$, we construct the OPS $\mathcal{G}_{\mathrm{d}} = \{G_{\mathrm{d}}^1 = G_{\mathrm{a}}^1, G_{\mathrm{d}}^2 = G_{\mathrm{a}}^2 \| G_{\mathrm{a}}^3 \| G_{\mathrm{a}}^4 \| H_{\mathrm{d}}\}$.

Now, for the specification models E^1, E^2, and E^3 designed in Σ (Fig. 3), let E_{d}^1, E_{d}^2, and E_{d}^3 be the respective versions in Δ, such that $E_{\mathrm{d}}^i = \Pi^{-1}(E^i)$, $i = 1, 2, 3$. Let also E_{d}^4 be originally designed in Δ, as shown in Fig. 3. By $\mathrm{LP}(\mathcal{G}_{\mathrm{d}}, E_{\mathrm{d}}^i) = G_{\mathrm{d}}{}_{loc}^i$, we construct $G_{\mathrm{d}}{}_{loc}^i$ to be associated with E_{d}^i in synthesis. For the particular example, it follows that $G_{\mathrm{d}}{}_{loc}^1 = G_{\mathrm{d}}^1 \| G_{\mathrm{d}}^2$ and $G_{\mathrm{d}}{}_{loc}^i = G_{\mathrm{d}}^2$, for $i = 2, 3, 4$. The number of states of the corresponding automata is shown in Table III.

In comparison to Table II, E_{d}^4 has now been modeled by a 2-states automaton, instead of 1.310. Besides, the nonconflict test preserves the complexity and results of the LMC-P, which has been our goal. However, using a distinguisher complexifies unnecessarily simple problems, as those for specifications 1, 2, and 3. The result is that obtaining $V_{\mathrm{d}}{}_{loc}^i$ tends to be more complex than synthesizing V_{loc}^i.

IV. EXPLOITING DISTINGUISHERS IN LMC

Synthesis results in Tables II and III are now combined in order to take parallel advantages from modeling and synthesis. For that, specifications are partitioned into the sets I^{Σ} and I^{Δ}, identifying, respectively, their alphabets. The distinguished LMC problem (DLMC-P), to be considered in this section, is introduced as follows.

Problem 4 [DLMC-P]: Let $G = \|_{j \in J} G^j$ be the monolithic plant model for an OPS $\mathcal{G}$ in Σ, and $D : \Sigma^* \to 2^{\Delta^*}$ be a predictable distinguisher. Given the sets of specifications $E^i \subseteq \Sigma^*$, $i \in I^{\Sigma}$ and $E_{\mathrm{d}}^i \subseteq \Delta^*$, $i \in I^{\Delta}$, defining a desired behavior $K = [\|_{i \in I^{\Sigma}} E^i \| L_m(G)] \| \Pi[\|_{i \in I^{\Delta}} E_{\mathrm{d}}^i \| L_m(G_{\mathrm{d}})]$, find nonblocking local supervisors S_{loc}^i for G_{loc}^i, $i \in I^{\Sigma}$, and $S_{\mathrm{d}}{}_{loc}^i$ for $G_{\mathrm{d}}{}_{loc}^i$, $i \in I^{\Delta}$, such that $S = (\wedge_{i \in I^{\Sigma}} S_{loc}^i) \wedge [\wedge_{i \in I^{\Delta}} \Pi(S_{\mathrm{d}}{}_{loc}^i)]$ is nonblocking with respect to G and $L_m(S/G) \subseteq K$.

The results to be presented next show the conditions under which a solution to DLMC-P optimally solves the original LMC-P (and indirectly the SC-P) and yet it takes advantages from distinguishers, when necessary.

A. Proper Local Modular Synthesis With Distinguishers

We show now that some plants can be separated from synthesis, without affecting the calculation of a controller.

Definition 6: For $i \in I^{\Sigma}$, let $E^i \subseteq \Sigma^*$ be a local specification and $\mathcal{G}$ be an OPS containing DES plants G^j designed in Σ^{G^j}, such that $G_{loc}^i = \mathrm{LP}(\mathcal{G}, E^i)$. Let also $G_{\mathrm{a}}^j = \Pi^{-1}(G^j)$ be the version of G^j in Δ, H_{d} be a distinguisher model, and C^i be such that $C^i = \{j \in J \mid \Sigma^{G^j} \cap \Sigma^{E^i} = \emptyset \wedge \Delta^{G_{\mathrm{a}}^j} \cap \Delta^{H_{\mathrm{d}}} \neq \emptyset\}$.

A complementary plant G_{c}^i, for G_{loc}^i, is such that

$$G_{\mathrm{c}}^i = \begin{cases} \|_{j' \in C^i} G^j & \text{if } C^i \neq \emptyset \\ H_{\Sigma}, & \text{otherwise.} \end{cases}$$

In words, G_{c}^i composes all plants G^j not in G_{loc}^i (no event-set intersection with E^i) but whose correspondent refined version (G_{a}^j) composes $G_{\mathrm{d}}{}_{loc}^i$ by the influence of H_{d} (if case). Otherwise, i.e., if both G_{loc}^i and $G_{\mathrm{d}}{}_{loc}^i$ compose the same plants, then there is no complementary plant and G_{c}^i is *neutral*.

Lemma 1 [19]: For a local specification $E^i \subseteq \Sigma^*$ and an OPS $\mathcal{G}$, let $G_{loc}^i = \mathrm{LP}(\mathcal{G}, E^i)$ and let G_{c}^i be the complementary plant for G_{loc}^i, as in Definition 6. Then, $\sup\mathcal{C}(K_{loc}^i, G_{loc}^i \| G_{\mathrm{c}}^i) = \sup\mathcal{C}(K_{loc}^i, G_{loc}^i) \| L_m(G_{\mathrm{c}}^i)$.

That is, plants that are not restricted by a given specification are not necessary to be considered in synthesis. This idea can be used to show an important relationship between LMC-P and its instance with distinguishers [15].

Lemma 2: For a local specification $E^i \subseteq \Sigma^*$ and an OPS $\mathcal{G}$ in Σ, let $G_{loc}^i = \mathrm{LP}(\mathcal{G}, E^i)$. For a local specification $E_{\mathrm{d}}^i \subseteq \Delta^*$ and an OPS $\mathcal{G}_{\mathrm{d}}$ in Δ, let $G_{\mathrm{d}}{}_{loc}^i = \mathrm{LP}(\mathcal{G}_{\mathrm{d}}, E_{\mathrm{d}}^i)$. Let also G_{c}^i be the complementary plant for G_{loc}^i, as in Definition 6. Then, $\Pi[\sup\mathcal{C}(K_{\mathrm{d}}{}_{loc}^i, G_{\mathrm{d}}{}_{loc}^i)] = \sup\mathcal{C}(K_{loc}^i, G_{loc}^i) \| L_m(G_{\mathrm{c}}^i)$.

That is, the LMC handled with or without distinguishers leads to the same result, and therefore, distinguishers can be used only when appropriate [16].

Theorem 1: Given the OPSs $\mathcal{G}$ and $\mathcal{G}_{\mathrm{d}}$, the set of specifications $E^i \subseteq \Sigma^*$, $i \in I^\Sigma$, and the set of specifications $E_{\mathrm{d}}^i \subseteq \Delta^*$, $i \in I^\Delta$, it follows that $\|^{i\in I} \sup\mathcal{C}(K_{loc}^i, G_{loc}^i) = [\|_{i\in I^\Sigma} \sup\mathcal{C}(K_{loc}^i, G_{loc}^i)] \ \| \ [\|_{i\in I^\Delta} \ \Pi(\sup\mathcal{C}(K_{\mathrm{d}}{}_{loc}^i, G_{\mathrm{d}}{}_{loc}^i))]$.

Theorem 1 partitions the set of specifications into two subsets for them to be solved with or without distinguishers. The resolution of any versions of the problem is equivalent, either locally or globally, and the following result follows.

Corollary 1: Let S^Σ be a set of supervisors S_{loc}^i, $i \in I$. Let $S^{\Sigma\Delta}$ be a set of supervisors S_{loc}^i, $i \in I^\Sigma$, and $S_{\mathrm{d}}{}_{loc}^i$, $i \in I^\Delta$. If S^Σ is an optimal solution to LMC-P, then $S^{\Sigma\Delta}$ is an optimal solution to DLMC-P.

That is, the use of distinguishers in LMC does not introduce or eliminate conflict on the resulting set of local modular supervisors. When the control problem is naturally conflict-free, the set $S^{\Sigma\Delta}$ is an optimal solution to DLMC-P and, as a consequence, also to LMC-P and SC-P.

V. Efficient Synthesis With Distinguishers

The results next are proposed to show how advantages from approximations in the DLMC-P can be taken.

For a set $\mathcal{GG}$ of automata $G^{j'}$, $j' \in J'$ modeling a DES plant with events in Σ, let $\mathcal{GG}_{\mathrm{a}}$ be a set of automata $G_{\mathrm{a}}^{j'}$, $j' \in J'$ such that $G_{\mathrm{a}}^{j'} = \Pi^{-1}(G^{j'})$. Let H_{da}^i be an automaton modeling, a distinguisher, such that $L(H_{\mathrm{da}}^i) = L_{\mathrm{da}} \supseteq L_{\mathrm{d}} = L(H_{\mathrm{d}})$, with $\Delta^{H_{\mathrm{d}}} = \Delta^{H_{\mathrm{da}}^i}$. Let also a set of specifications $E^i \subseteq \Sigma^*$, $i \in I^\Sigma$, and a set of specifications $E_{\mathrm{d}}^i \subseteq \Delta^*$, $i \in I^\Delta$. For a distinguished local plant $G_{\mathrm{d}}{}_{loc}^i = \mathrm{LP}[\mathrm{async}(\{\mathcal{GG}_{\mathrm{a}}, H_{\mathrm{d}}\}), E_{\mathrm{d}}^i]$, define a *local D-approximation* for $G_{\mathrm{d}}{}_{loc}^i$ as the automaton $G_{\mathrm{a}}{}_{loc}^i = \mathrm{LP}[\mathrm{async}(\{\mathcal{GG}_{\mathrm{a}}, H_{\mathrm{da}}^i\}), E_{\mathrm{d}}^i]$.

From $G_{\mathrm{a}}{}_{loc}^i$, each module i of the LMC with distinguishers is prepared to be associated with H_{da}^i, a particular case of a distinguisher H_{d} that "distinguishes less" than H_{d} (as $L_{\mathrm{d}} \subseteq L_{\mathrm{da}}$), but it is such that it includes L_{d}. When associated with the set $\mathcal{GG}_{\mathrm{a}}$ of plant models, H_{da}^i produces a *D-approximation* $G_{\mathrm{a}}{}_{loc}^i$ (as in Definition 4) to the plant $G_{\mathrm{d}}{}_{loc}^i$.

For $G_{\mathrm{a}}^{j'}$, $G_{\mathrm{d}}{}_{loc}^i$, and $G_{\mathrm{a}}{}_{loc}^i$, it follows that $\Delta^{G_{\mathrm{a}}^{j'}} \cap \Delta^{G_{\mathrm{d}}{}_{loc}^i} \neq \emptyset$ implies $\Delta^{G_{\mathrm{a}}^{j'}} \cap \Delta^{G_{\mathrm{a}}{}_{loc}^i} \neq \emptyset$, as $\Delta^{H_{\mathrm{d}}} = \Delta^{H_{\mathrm{da}}^i}$ by assumption. That is, $G_{\mathrm{d}}{}_{loc}^i$ and $G_{\mathrm{a}}{}_{loc}^i$ locally compose the same plants. Even so, $L_{\mathrm{d}} \subseteq L_{\mathrm{da}}$ suggests that H_{da}^i can be modeled by an automaton simpler than H_{d}, in terms of number of states, which makes $G_{\mathrm{a}}{}_{loc}^i$ also simpler than $G_{\mathrm{d}}{}_{loc}^i$.

The effect of replacing $G_{\mathrm{d}}{}_{loc}^i$ in synthesis by an approximation $G_{\mathrm{a}}{}_{loc}^i$ can be formalized by a simple application of the Proposition 1 to a local case, i.e., if $\sup\mathcal{C}(K_{\mathrm{a}}{}_{loc}^i, G_{\mathrm{a}}{}_{loc}^i) = \sup\mathcal{P}_{\Sigma_u}[K_{\mathrm{a}}{}_{loc}^i, L(G_{\mathrm{a}}{}_{loc}^i)]$ then $\sup\mathcal{C}(K_{\mathrm{a}}{}_{loc}^i, G_{\mathrm{a}}{}_{loc}^i) \cap L_{\mathrm{d}} = \sup\mathcal{C}(K_{\mathrm{d}}{}_{loc}^i, G_{\mathrm{d}}{}_{loc}^i)$, and therefore, by Lemma 2, $\Pi[\sup\mathcal{C}(K_{\mathrm{a}}{}_{loc}^i, G_{\mathrm{a}}{}_{loc}^i) \cap L_{\mathrm{d}}] = \sup\mathcal{C}(K_{loc}^i, G_{loc}^i) \| L_m(G_{\mathrm{c}}^i)$.

Note that a local D-approximation can variate from the coarsest $G_{\mathrm{a}}{}_{loc}^i$, constructed using a distinguisher H_{da}^i, such that $L(H_{\mathrm{da}}^i) = L_{\mathrm{da}} = L(H_\Delta) = \Delta^*$, to $G_{\mathrm{d}}{}_{loc}^i$ constructed using a H_{da}^i such that $L(H_{\mathrm{da}}^i) = L_{\mathrm{da}} = L_{\mathrm{d}} = L(H_{\mathrm{d}})$. It is expected that, satisfied the conditions of the Proposition 1, the resulting control solution is preserved independently of which D-approximation is used. This suggests that each local problem handled in Δ could be approached using a particular D-approximation, as convenient. Theorem 2 formalizes this idea.

Theorem 2: For a set of specifications $E^i \subseteq \Sigma^*$, $i \in I^\Sigma$, and a set of specifications $E_{\mathrm{d}}^i \subseteq \Delta^*$, $i \in I^\Delta$, if the conditions of Proposition 1 are satisfied, then it follows that $\|_{i\in I} \sup\mathcal{C}(K_{loc}^i, G_{loc}^i) = [\|_{i\in I^\Sigma} \sup\mathcal{C}(K_{loc}^i, G_{loc}^i)] \ \| \ [\|_{i\in I^\Delta} \ \Pi(\sup\mathcal{C}(K_{\mathrm{a}}{}_{loc}^i, G_{\mathrm{a}}{}_{loc}^i) \| L_{\mathrm{d}})]$.

In other words, under the conditions of Proposition 1, any combination of local solutions in Σ or in Δ solves the original LMC-P. Note that $\sup\mathcal{C}(K_{\mathrm{d}}{}_{loc}^i, G_{\mathrm{d}}{}_{loc}^i)$ is a particular (predictable) instance of $\sup\mathcal{C}(K_{\mathrm{a}}{}_{loc}^i, G_{\mathrm{a}}{}_{loc}^i)$, for which a distinguisher H_{da}^i, with $L(H_{\mathrm{da}}^i) = L_{\mathrm{da}} = L_{\mathrm{d}} = L(H_{\mathrm{d}})$, has been used.

A. Example Revisited

Following the idea of Theorem 2, we split the set of specifications $I = \{1, \cdots, 4\}$ into the subsets $I^\Sigma = \{1, 2, 3\} \subset I$ and $I^\Delta = \{4\} \subset I$. Solutions to problems in I^Σ and I^Δ have been, respectively, presented in Tables II and III.

By using distinguishers to obtain only $V_{\mathrm{d}}{}_{loc}^4$, we avoid to complexify problems 1, 2, and 3, taking parallel advantages to model E_{d}^4. One can perform the nonconflict test by projecting $V_{\mathrm{d}}{}_{loc}^4$ into Σ. The composition $(\|_{i=1,\cdots,3} V_{loc}^i) \ \| \ [\Pi(V_{\mathrm{d}}{}_{loc}^4)]$ is a blocking automaton with 276 states, as expected.

For problem 4, which has been handled in I^Δ, consider now using an approximation for the plant $G_{\mathrm{d}}{}_{loc}^4$. We first use the coarsest approximation $G_{\mathrm{ca}}{}_{loc}^4$, which is constructed by entirely removing distinctions from $G_{\mathrm{d}}{}_{loc}^4$. To produce such effect, L_{dca}^4 has been modeled by an automaton $H_{\mathrm{dca}}^4 = H_\Delta$, with $\Delta^{H_{\mathrm{d}}} = \Delta^{H_{\mathrm{dca}}^4}$, which makes $G_{\mathrm{ca}}{}_{loc}^4$ a local D-approximation for $G_{\mathrm{d}}{}_{loc}^4$. Then, the local plant is now given by $G_{\mathrm{ca}}{}_{loc}^4 = \mathrm{LP}[\mathrm{async}(\{\mathcal{GG}_{\mathrm{a}}, H_{\mathrm{dca}}^4\}), E_{\mathrm{d}}^4]$.

The specification is modeled by $K_{\mathrm{ca}}{}_{loc}^4$ representing $L_m(G_{\mathrm{ca}}{}_{loc}^4) \cap E_{\mathrm{d}}^4$, which leads to a supervisor $V_{\mathrm{ca}}{}_{loc}^4$ that models $\sup\mathcal{C}(K_{\mathrm{ca}}{}_{loc}^4, G_{\mathrm{ca}}{}_{loc}^4)$, as illustrated in Table IV (first row).

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV

SOLUTIONS TO PROBLEM 4 OBTAINED WITH APPROXIMATIONS

$G{\mathrm{ca}}^4_{loc}$	E^4_{d}	$K{\mathrm{ca}}^4_{loc}$	$V{\mathrm{ca}}^4_{loc}$	$V{\mathrm{ca}}^4_{loc} \| H_{\mathrm{d}}$
12	2	24	18	1.878
Nonconflict test: $(\|_{i \in I^{\Sigma}} V^i_{loc}) \parallel (\Pi(V{\mathrm{ca}}^4_{loc} \| H_{\mathrm{d}}))$				276
$G{\mathrm{a}}^4_{loc}$	E^i_{d}	$K{\mathrm{a}}^i_{loc}$	$V{\mathrm{a}}^i_{loc}$	$V{\mathrm{a}}^i_{loc} \| H_{\mathrm{d}}$
18	2	36	30	2.400
Nonconflict test: $(\|_{i \in I^{\Sigma}} V^i_{loc}) \parallel (\Pi(V{\mathrm{a}}^4_{loc} \| H_{\mathrm{d}}))$				276

To synthesize $V{\mathrm{ca}}^4_{loc}$, it has been used as an automaton with only 24 states, instead of 2.662 as when obtaining $V{\mathrm{d}}^4_{loc}$. As $V{\mathrm{ca}}^4_{loc}$ is controllable and it is also nonconflicting with respect to H_{d}, it can be used to implement a local solution to local problem number 4 (by Proposition 1). Moreover, the global nonconflicting condition (second row) remains unchanged (276 states), as expected. However, it can be shown that $\sup\mathcal{C}(K{\mathrm{ca}}^4_{loc}, G{\mathrm{ca}}^4_{loc}) \neq \sup\mathcal{P}_{\Sigma_{\mathrm{u}}}[K{\mathrm{ca}}^4_{loc}, L(G{\mathrm{ca}}^4_{loc})]$. In fact, while $V{\mathrm{ca}}^4_{loc}$ has 18 states, the automaton for $\sup\mathcal{P}_{\Sigma_{\mathrm{u}}}[K{\mathrm{ca}}^4_{loc}, L(G{\mathrm{ca}}^4_{loc})]$ has 30 states.

In order to synthesize an optimal solution, we compose $H{\mathrm{d}}^2_f$ to H^4_{dca} (see Section III). Let, then an improved approximation for $G{\mathrm{d}}^4_{loc}$ be now given by $G{\mathrm{a}}^4_{loc} = \mathrm{LP}(\mathrm{async}[\{\mathcal{G}\mathcal{G}_{\mathrm{a}}, H^4_{\mathrm{da}} = H_{\Delta} \| H{\mathrm{d}}^2_f\}], E^4_{\mathrm{d}})$. The specification is now modeled by $K{\mathrm{a}}^4_{loc}$ representing $L_m(G{\mathrm{a}}^4_{loc}) \cap E^4_{\mathrm{d}}$, which leads to a supervisor $V{\mathrm{a}}^4_{loc}$ that models $\sup\mathcal{C}(K{\mathrm{a}}^4_{loc}, G{\mathrm{a}}^4_{loc})$ [Table IV (third row)].

Now, $V{\mathrm{a}}^4_{loc}$ is obtained from an automaton with 36 states, also much smaller than the one used to compute $V{\mathrm{d}}^4_{loc}$, with 2.662 states. The global nonconflicting condition (last row) remains still unchanged (276 states), as expected. By the nonconflict of $\sup\mathcal{C}(K{\mathrm{a}}^4_{loc}, G{\mathrm{a}}^4_{loc})$ and L_{d}, it follows that $\sup\mathcal{C}(K{\mathrm{a}}^4_{loc}, G{\mathrm{a}}^4_{loc}) \cap L_{\mathrm{d}}$ is a local solution to the DLMC-P (by Proposition 1). Moreover, now it also follows that $\sup\mathcal{C}(K{\mathrm{a}}^4_{loc}, G{\mathrm{a}}^4_{loc}) = \sup\mathcal{P}_{\Sigma_u}[K{\mathrm{a}}^4_{loc}, L(G{\mathrm{a}}^4_{loc})]$, which is modeled by a 30-state automaton. By Proposition 1, $V{\mathrm{a}}^4_{loc}$ and H_{d} can be used to implement a supervision system $\sup\mathcal{C}(K{\mathrm{a}}^4_{loc}, G{\mathrm{a}}^4_{loc}) \cap L_{\mathrm{d}} = \sup\mathcal{C}(K{\mathrm{d}}^4_{loc}, G{\mathrm{d}}^4_{loc})$. By Theorem 2, $V{\mathrm{a}}^4_{loc} \| H_{\mathrm{d}}$ and V^i_{loc}, $i = 1, 2, 3$ can be combined to solve the LMC-P.

We remark that to obtain a supervisor $V{\mathrm{a}}^4_{loc}$ that leads to solve the LMC-P there is no need to explicitly compose $V{\mathrm{a}}^4_{loc} \| H_{\mathrm{d}}$. The controller $V{\mathrm{a}}^4_{loc} \| H_{\mathrm{d}}$ might be instead implemented by two concurrent automata $V{\mathrm{a}}^4_{loc}$ and H_{d} such that the possibly expensive composition is never actually computed.

VI. CONCLUSION

This paper combines LMC, distinguishers, and approximations to solve DES control problems. The combination allows reducing both modeling and synthesis effort, contributing for the SCT to be used to handle complex large-scale systems.

Although the synthesis procedure is conducted locally, we remark that a nonconflict test is required on local supervisors and distinguisher, in order to ensure global nonblocking and optimality. As future research, we aim to investigate how such test could be simplified. We also estimate that approximations could be constructed such that they could lead to least restrictiveness and nonconflicting supervisors by construction.

REFERENCES

[1] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2008.

[3] M. Teixeira, R. Malik, J. E. R. Cury, and M. H. de Queiroz, "Supervisory control of DES with extended finite-state machines and variable abstraction," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 118–129, Jan. 2015.

[4] J. E. Cury, M. H. de Queiroz, G. Bouzon, and M. Teixeira, "Supervisory control of discrete event systems with distinguishers," *Automatica*, vol. 56, pp. 93–104, Jun. 2015.

[5] A. E. C. da Cunha and J. E. R. Cury, "Hierarchical supervisory control based on discrete event systems with flexible marking," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2242–2253, Dec. 2007.

[6] K. Schmidt, T. Moor, and S. Perk, "Nonblocking hierarchical control of decentralized discrete event systems," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2252–2265, Nov. 2008.

[7] K. C. Wong and W. M. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 6, no. 3, pp. 241–273, 1996.

[8] T. Le Gall, B. Jeannet, and H. Marchand, "Supervisory control of infinite symbolic systems using abstract interpretation," in *Proc. Int. Conf. Decision Control*, Seville, Spain, 2005, pp. 30–35.

[9] L. Ouedraogo, R. Kumar, R. Malik, and K. Åkesson, "Nonblocking and safe control of discrete-event systems modeled as extended finite automata," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 560–569, Jul. 2011.

[10] Z. Fei, S. Reveliotis, S. Miremadi, and K. Åkesson, "A BDD-based approach for designing maximally permissive deadlock avoidance policies for complex resource allocation systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 990–1006, Jul. 2015.

[11] M. R. Shoaei, L. Feng, and B. Lennartson, "Abstractions for nonblocking supervisory control of extended finite automata," in *Proc. Int. Conf. Autom. Sci. Eng.*, Seoul, South Korea, Aug. 2012, pp. 364–370.

[12] S. Mohajerani, R. Malik, and M. Fabian, "Compositional synthesis of supervisors in the form of state machines and state maps," *Automatica*, vol. 76, pp. 277–281, Feb. 2017.

[13] M. H. D. Queiroz and J. E. R. Cury, "Modular supervisory control of large scale discrete event systems," in *Proc. Int. Workshop Discrete Event Syst.*, Ghent, Belgium, 2000, pp. 103–110.

[14] G. Bouzon, M. H. de Queiroz, and J. E. R. Cury, "Supervisory control of DES with distinguishing sensors," in *Proc. Int. Workshop Discrete Event Syst.*, Gothenburg, Sweden, 2008, pp. 390–391.

[15] M. Teixeira, J. E. R. Cury, and M. H. de Queiroz, "Local modular supervisory control of DES with distinguishers," in *Proc. Int. Conf. Emerg. Tech. Factory Autom.*, Toulouse, France, Sep. 2011, pp. 1–8.

[16] M. Teixeira, J. E. R. Cury, and M. H. de Queiroz, "Local modular control with distinguishers applied to a manufacturing system," *IFAC Proc. Volumes*, vol. 46, no. 9, pp. 263–268, 2013.

[17] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[18] R. J. Leduc, M. Lawford, and P. Dai, "Hierarchical interface-based supervisory control of a flexible manufacturing system," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 654–668, Jul. 2006.

[19] M. H. D. Queiroz and J. E. R. Cury, "Modular control of composed systems," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, Jun. 2000, pp. 4051–4055.

[20] W. M. Wonham and P. J. Ramadge, "Modular supervisory control of discrete-event systems," *Math. Control, Signals, Syst.*, vol. 1, no. 1, pp. 13–30, 1988.

[21] K. C. Wong and W. M. Wonham, "Modular control and coordination of discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 8, no. 3, pp. 247–297, 1998.

[22] R. S. S. Aguiar, A. E. C. Cunha, J. E. R. Cury, and M. H. Queiroz, "Heuristic search of supervisors by approximated distinguishers," *IFAC Proc. Volumes*, vol. 46, no. 22, pp. 121–126, 2013.

[23] M. Teixeira, R. Malik, J. E. R. Cury, and M. H. de Queiroz, "Variable abstraction and approximations in supervisory control synthesis," in *Proc. Amer. Control Conf.*, Washington, DC, USA, Jun. 2013, pp. 132–137.

[24] R. Malik and M. Teixeira, "Modular supervisor synthesis for extended finite-state machines subject to controllability," in *Proc. Int. Workshop Discrete Event Syst.*, May/Jun. 2016, pp. 91–96.

[25] M. H. de Queiroz and J. E. R. Cury, "Modular multitasking supervisory control of composite discrete-event systems," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 91–96, 2005.