Local Refinement of l-complete Approximations for Supervisory Control of Hybrid Systems ⋆

Jung–Min Yang * **Thomas Moor** ** **Jörg Raisch** ***,****

* *School of Electronics Engineering, Kyungpook National University, 80 Daehakro, Bukgu, Daegu, 41566, Republic of Korea (e-mail: jmyang@ee.knu.ac.kr)*
** *Lehrstuhl für Regelungstechnik at Friedrich-Alexander Universität Erlangen-Nürnberg, D-91058 Erlangen, Germany (e-mail: lrt@fau.de)*
*** *Fachgebiet Regelungssysteme, Technische Universitaet Berlin, 10587 Berlin, Germany*
**** *Systems and Control Theory Group, Max Planck Institute for Dynamics of Complex Technical Systems, D-39106 Magdeburg, Germany (e-mail: raisch@control.tu-berlin.de)*

Abstract: l-complete approximation is a discrete abstraction method for a specific class of hybrid control problems involving purely discrete specifications. It allows for global refinement if the subsequent synthesis of supervisory control should fail for the currently selected abstraction level. In this paper, we present a methodology of local refinement for l-complete approximations. If the strongest l-complete approximation of a given hybrid system does not guarantee the existence of a suitable supervisor for a given specification, the proposed scheme refines the abstract model only in a local set of states that violate a controllability condition. Compared to the standard unfocused and global refinement procedure, this may significantly reduce the computational burden both in the abstraction step and the subsequent controller synthesis step.

Keywords: Hybrid systems, l-complete approximations, local refinement, discrete event systems, supervisory control.

1. INTRODUCTION

The analysis and control of hybrid systems has become an important subject in modern control theory. A requisite to successful controller synthesis for hybrid systems is a modeling formalism that describes the dynamics of hybrid systems on a level of abstraction such that an appropriate controller can be designed to meet a desirable specification. A number of mathematical models for hybrid systems have been reported for this purpose in various frameworks; see, e.g., Alur, Henzinger, Lafferriere, & Pappas (2000); Girard & Pappas (2009); Tabuada (2009); Tarraf (2014) and the references cited therein.

This paper addresses the local refinement of l-complete approximations (Moor & Raisch, 1999; Moor, Raisch, & O'Young, 2002), a discrete abstraction inspired by Willems' behavioral systems theory (e.g., Willems (1991)). l-complete approximation has been suggested as an abstraction method for specific hybrid control problems where the plant to be controlled exhibits discrete-valued (symbolic) control input and output signals, and is subject to an inclusion-type specification in these signals. In standard l-complete approximations, a finite state machine with a certain level of abstraction quantified by the integer parameter l is induced by the external behavior of the hybrid system. A variant of Ramadge and Wonham's supervisory control theory (e.g., Cassandras & Lafortune (2008); Ramadge & Wonham (1987, 1989)) is subsequently applied to the approximated model to enforce a given inclusion-type specification for the approximation. It is shown in Moor & Raisch (1999); Moor et al. (2002) that if the supervisor suitably restricts the l-complete approximation behavior, it also accomplishes the control objective for the underlying hybrid system. Recent extensions have been reported in Park & Raisch (2015); Schmuck & Raisch (2014). In Schmuck & Raisch (2014), the notion of asynchronous l-complete approximations was introduced, and Park & Raisch (2015) presents a controller synthesis procedure for l-complete approximation models with a partially observed output set.

The existence of an appropriate supervisor for a given specification depends on the approximation accuracy of the l-complete approximation, namely, if the abstract model is too *coarse*, no supervisor may exist that meets the specification. In Moor & Raisch (1999); Moor et al. (2002); Park & Raisch (2015); Schmuck & Raisch (2014), the only method to resolve this conflict is to construct a globally *refined* model, i.e., to increase the integer parameter l. This will improve approximation accuracy, but at the

⋆ The work of J.–M. Yang was supported in part by Researcher Exchange Program under Memorandum of Understanding between the NRF–DAAD and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2015R1D1A1A01056764).

cost of significant computational complexity. Furthermore, the existence or non-existence of a suitable supervisor is revealed only after the refinement step and the subsequent controller synthesis procedure have been entirely carried out.

Motivated by this drawback, we propose local refinement of l-complete approximations. In our scheme, even if no supervisor is found to exist for the current abstraction, global refinement of the model is not undertaken. Instead, we refine only the part of the current abstraction violating a certain controllability condition. This information is available from the (failed) controller synthesis step. Since the refinement range is local, the part of the previously generated supervisor associated with the unrefined part of the model can be reused, hence allowing for computationally efficient derivation of the supervisor.

A study on the local refinement of abstractions in the context of controller synthesis that also applies to l-complete approximations is provided by (Moor, Davoren, & Raisch, 2006). There, the authors utilize the more general concept of *exhaustive experiments* as a basis for abstractions and state their discussion exclusively in terms of behaviors. In contrast, the present paper exploits the special case of l-complete approximations and presents a refinement scheme explicitly in terms of state machines. Other refinement methodologies for verification and control of hybrid systems are found, e.g., in Clarke, Fehnker, Han, Krogh, Ouaknine, Stursberg, & Theobald (2003); Stursberg (2006). The approaches of Clarke et al. (2003); Stursberg (2006) are similar in spirit to the proposed scheme since both try to induce the refined model while avoiding exhaustive model checking and refinement. Unlike Clarke et al. (2003); Stursberg (2006), however, our scheme does not need any counter example or candidate path to determine the validity of the evolved model. Only the composite finite state machine obtained in the previous controller synthesis procedure will be utilized to derive a more detailed system abstraction.

We first summarize the concept of strongest l-complete approximations as discrete abstractions for continuous or hybrid systems and the synthesis of supervisory control for such abstractions. We then propose a procedure for local refinement of a state machine realization of a given l-complete approximation, where the state set and state transition relation are extended to compensate for the failure of implementing adequate supervisory control. A simple example will be used through the entire discussion, illustrating the suggested idea and demonstrating the applicability of the proposed strategy.

2. L-COMPLETE APPROXIMATIONS OF HYBRID SYSTEMS

In Willems' behavioral framework, a dynamical system is defined as a triple $\Sigma = (T, W, \mathcal{B})$, where T is the time axis, W is the external signal space, and $\mathcal{B} \subseteq W^T := \{\mathbf{w} | \mathbf{w} : T \to W\}$ is the behavior, or the set of all external system signals evolving on T and taking values in W. Since we focus on Σ with discrete behaviors, let us assume that $T = \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $|W| \in \mathbb{N}$. In the context of control systems, $W = U \times Y$, where U and Y are the input and output set, respectively ($|U|, |Y| \in \mathbb{N}$).

For algorithmic purposes, we use state machines as realizations of dynamical systems. A state machine is a quadruple $P = (X, W, \delta, X_0)$, where X is the state set, $W = U \times Y$ is the external signal space, $\delta \subseteq X \times W \times X$ is the next state relation, and $X_0 \subseteq X$ is the set of initial states. P is called a finite state machine if $|X| \in \mathbb{N}$. P is called *past-induced* if there is no more than one initial state and if for every reachable state x and every symbol $\omega \in W$ there is at most one successor state x' such that $(x, \omega, x') \in \delta$; in automata theory, this is also referred to as *determinism*. P induces the *full behavior* $\mathcal{B}_s := \{(\mathbf{w}, \mathbf{x}) | (\mathbf{x}(t), \mathbf{w}(t), \mathbf{x}(t+1)) \in \delta \ \forall t \in \mathbb{N}_0, \mathbf{x}(0) \in X_0\}$ and the state space system $\Sigma_s := (\mathbb{N}_0, W \times X, \mathcal{B}_s)$. The *external behavior* $\mathcal{B}$ of Σ_s is the projection of $\mathcal{B}_s$ onto $W^{\mathbb{N}_0}$, i.e., $\mathcal{B} := \mathscr{P}_W \mathcal{B}_s = \{\mathbf{w} | \exists \mathbf{x} \text{ s.t. } (\mathbf{w}, \mathbf{x}) \in \mathcal{B}_s\}$. If a state machine P' induces the external behavior $\mathcal{B}$ of a system $\Sigma' = (\mathbb{N}_0, W, \mathcal{B})$, P' is termed a realization of Σ', denoted by $\Sigma' \cong P'$.

Let σ^t denote the backwards t-shift, i.e., $(\sigma^t \mathbf{w})(\tau) := \mathbf{w}(t+\tau) \ \forall \tau \in \mathbb{N}_0$ and $\sigma := \sigma^1$. In the following, we will focus on the case where the behavior is shift invariant, i.e., $\sigma\mathcal{B} = \mathcal{B}$, where $\sigma\mathcal{B}$ is shorthand for $\{\sigma\mathbf{w} | \mathbf{w} \in \mathcal{B}\}$. This implies time-invariance of the corresponding system $\Sigma = (T, W, \mathcal{B})$ (e.g., Willems (1991)). Let the system $\Sigma = (\mathbb{N}_0, W, \mathcal{B})$ with $|W| \in \mathbb{N}$ be realized by a hybrid state machine $P = (X, W, \delta, X_0)$ with $X \subseteq \mathbb{R}^n \times D, |D| \in \mathbb{N}$, namely, P may have infinitely many states. In addition, we assume that P is an *I/S/O machine*, i.e., $\forall x \in X$ and $u \in U$, there uniquely exist $y \in Y$ and $x' \in X$ such that $(x, (u, y), x') \in \delta$ (Moor & Raisch, 1999); this is a different form of determinism that must not be confused with past-inducedness.

We define $\mathbf{w}|_{[t_1,t_2]}$ and $\mathcal{B}|_{[t_1,t_2]}$ $(t_1 \leqslant t_2)$ as the restriction of $\mathbf{w}$ and $\mathcal{B}$ to the domain $[t_1, t_2] \cap \mathbb{N}_0$, respectively. In behavioral systems theory (e.g., Willems (1991)), a dynamical system is called *l-complete* if its behavior can be fully described by local properties of the system evaluated on time intervals of length $l+1$, $l \in \mathbb{N}$. In formal terms, a shift invariant system $\Sigma = (T, W, \mathcal{B})$ is l-complete if

$$\mathbf{w} \in \mathcal{B} \Leftrightarrow (\sigma^t \mathbf{w})|_{[0,l]} \in \mathcal{B}|_{[0,l]} \ \forall t \in \mathbb{N}_0. \tag{1}$$

In Moor et al. (2002), a formalism is presented to realize an l-complete hybrid system with finite external signal space by a past-induced finite state machine as follows.

Theorem 1. (Moor & Raisch, 1999) Given an l-complete shift invariant hybrid system $\Sigma = (T, W, \mathcal{B})$, let $Z_l := \{\omega^*\} \cup_{1 \leqslant r \leqslant l} W^r$ and $Z_0 := \{\omega^*\}$, where $\omega^* \notin W$ is a dummy character meaning "no external signal present so far." Let $\delta_l := \cup_{0 \leqslant r \leqslant l} \delta_l^r \subseteq Z_l \times W \times Z_l$ where

$$\begin{aligned}
&\delta_l^0 := \{(\omega^*, \omega_0, \omega_0) | \langle \omega_0 \rangle \in \mathscr{B}|_{[0,0]}\}, \\
&\delta_l^r := \{(\langle \omega_0, \ldots, \omega_{r-1} \rangle, \omega_r, \langle \omega_0, \ldots, \omega_r \rangle) | \\
&\quad \langle \omega_0, \ldots, \omega_r \rangle \in \mathscr{B}|_{[0,r]}\}, \ 1 \leqslant r < l, \\
&\delta_l^l := \\
&\quad \{(\langle \omega_0, \ldots, \omega_{l-1} \rangle, \omega_l, \langle \omega_1, \ldots, \omega_l \rangle) | \langle \omega_0, \ldots, \omega_l \rangle \in \mathscr{B}|_{[0,l]}\}.
\end{aligned} \tag{2}$$

Then, $P_l := (Z_l, W, \delta_l, Z_0)$ is a past-induced realization of Σ.

If $\Sigma = (T, W, \mathcal{B})$ is not l-complete, one needs to find the *strongest l-complete approximation* $\Sigma_l = (T, W, \mathcal{B}_l)$ of Σ. It is defined by the following properties (Moor & Raisch, 1999):

(i) Σ_l is l-complete;
(ii) $\mathcal{B}_l \supseteq \mathcal{B}$; and
(iii) if $\Sigma' = (T, W, \mathcal{B}')$ with $\mathcal{B}' \supseteq \mathcal{B}$ is also l-complete, then $\mathcal{B}_l \subseteq \mathcal{B}'$.

As stated in Theorem 1, an l-complete system Σ with finite external signal space can be realized by a finite state machine. Hence it is suitable to apply supervisory control theory to Σ. The motivation for defining Σ_l is to deal with the case that Σ is not l-complete. Instead of Σ, Σ_l will be used in supervisor synthesis. In Moor et al. (2002), it is shown that a valid supervisor for Σ_l is also a valid supervisor for Σ.

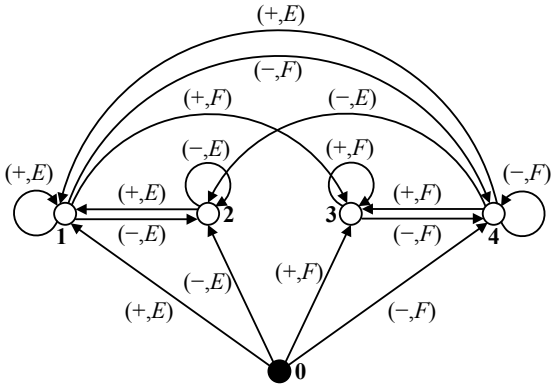

Fig. 1. Realization P_1 for 1-complete approximation Σ_1.

Example 2. Consider the water tank system with height 30 cm discussed in Moor et al. (2002); Park & Raisch (2015). Operating with a fixed sampling rate, the attached pump provides water in response to the discrete input $u(t)$. If $u(t)$="+", the pump adds water to the tank, increasing the water level $x(t)$ by 10 cm in the next sampling interval (unless the tank is full); else if $u(t)$="−", the pump removes water from the tank, decreasing the water level by 10 cm during the next sampling interval (unless the tank is empty). Hence, the range of $x(t)$ is always bounded by $0 \leqslant x(t) \leqslant 30$ in the example. The output $y(t)$ can take only two measurement values: $y(t) = E$ if $0 \leqslant x(t) \leqslant 15$, and $y(t) = F$ if $15 < x(t) \leqslant 30$. Let the water tank model be $\Sigma = (T, W, \mathcal{B})$ where $W = U \times Y$, $U = \{+, -\}$, and $Y = \{E, F\}$. Then Σ can be realized by a hybrid state machine $P = (X, W, \delta, X_0)$ where $X = X_0 = [0, 30]$, $w(t) = (u(t), y(t))$, and $(x(t), w(t), x(t+1)) \in \delta$ if $x(t+1) = f(x(t), u(t))$

$$:= \begin{cases} x(t) + 10 & u(t) = \text{“}+\text{”} \text{ and } 0 \leqslant x(t) \leqslant 20 \\ 30 & u(t) = \text{“}+\text{”} \text{ and } 20 < x(t) \leqslant 30 \\ x(t) - 10 & u(t) = \text{“}-\text{”} \text{ and } 10 < x(t) \leqslant 30 \\ 0 & u(t) = \text{“}-\text{”} \text{ and } 0 \leqslant x(t) \leqslant 10 \end{cases} \tag{3}$$

$$y(t) = g(x(t)) := \begin{cases} F & 15 < x(t) \leqslant 30 \\ E & 0 \leqslant x(t) \leqslant 15 \end{cases}$$

Fig. 1 shows the realization P_1 for the strongest 1-complete approximation Σ_1 of Σ. Since the parameter l is 1, the number of states of P_1 is $|U| \cdot |Y| + 1$. In this example, all states, abbreviated by $0, \ldots, 4$ in Fig. 1, can indeed be reached. For instance, a pair of external symbols $(+, E)$ followed by $(+, E)$ will drive P_1 into state $1 = (+, E)$. A detailed algorithm for deriving $\mathcal{B}_l$ and P_l is found in Moor et al. (2002).

3. SUPERVISORY CONTROL

To address supervisory control for hybrid systems, we recall relevant terminology and operations from Moor et al. (2002). Let $P_a = (A, W, \alpha, A_0)$ and $P_b = (B, W, \beta, B_0)$ be two state machines having the same external signal space W. P_a is *reachable* if every state $a \in A$ is reachable, namely a can be reached from a state $a_0 \in A_0$ through a chain of transitions of α. P_a is *temporally nonblocking* if for every reachable $a \in A$, $w \in W$ exists such that $(a, w, a') \in \alpha$ for some $a' \in A$. The latter is abbreviated by $(a, w, -) \in \alpha$. Similarly, $(a, w, a') \notin \alpha$ for all $a' \in A$ is abbreviated by $(a, w, -) \notin \alpha$. If $(a, w, -) \notin \alpha$, $\forall w \in W$, a is a *blocking state*. The *parallel composition* of P_a and P_b is defined as $P_a || P_b := (A \times B, W, \lambda, A_0 \times B_0)$ where $((a, b), w, (a', b')) \in \lambda \Leftrightarrow (a, w, a') \in \alpha$ and $(b, w, b') \in \beta$. P_a is a *substructure* of P_b, denoted by $P_a \subseteq P_b$, if $A \subseteq B$, $\alpha \subseteq \beta$, and $A_0 \subseteq B_0$. $P_a = (\varnothing, W, \varnothing, \varnothing) \subseteq P_b$ is the trivial substructure of P_b and is simply denoted by $P_a = \varnothing$. The *union* of P_a and P_b is defined as $P_a \cup P_b := (A \cup B, W, \alpha \cup \beta, A_0 \cup B_0)$.

Assume that for a hybrid system $\Sigma = (\mathbb{N}_0, W, \mathcal{B})$, $P_l = (Z_l, W, \delta_l, Z_0)$ is derived for a specific choice of l and a specification $\Sigma_{spec} = (\mathbb{N}_0, W, \mathcal{B}_{spec})$ is given with past-induced finite state realization $P_{spec} = (X_{spec}, W, \delta_{spec}, X_{spec0})$. The control problem on the approximation level is to determine the existence of a non-trivial supervisor such that the closed-loop formed by P_l and the supervisor (i) exhibits a behavior that is contained in $\mathcal{B}_{spec}$, (ii) is temporally nonblocking, and (iii) satisfies a controllability condition outlined below.

Supervisor synthesis is conducted in the following steps. First, we remove unacceptable trajectories from Σ_l by intersecting $\mathcal{B}_l$ and $\mathcal{B}_{spec}$. A realization of the intersection system $(\mathbb{N}_0, W, \mathcal{B}_l \cap \mathcal{B}_{spec})$ is $P_l || P_{spec} := (Q, W, \lambda, Q_0)$ where $Q = Z_l \times X_{spec}$, $\lambda \subseteq Q \times W \times Q$, and $Q_0 = Z_0 \times X_{spec0}$. Since unreachable states do not affect the induced behavior, we assume hereafter that $P_l || P_{spec}$ is the parallel composition obtained after removing all the unreachable states. Next, $P_l || P_{spec}$ is truncated so as to ensure the principle of supervisory control that the supervisor may only disable the input component in an explicit manner. To be more specific, if the supervisor disables a symbol $\omega \in W$ at a state in $P_l || P_{spec}$, it must disable all other symbols that have the same input component as ω, regardless of the difference in their output components. The following definition (Moor et al., 2002) epitomizes this restriction in formal terms.

Definition 3. Transitions $\delta^1 = (z_1, \omega_1, z_1') \in \delta_l$ and $\delta^2 = (z_2, \omega_2, z_2') \in \delta_l$ of P_l are *partners* if $z_1 = z_2$ and $\mathscr{P}_U \omega_1 = \mathscr{P}_U \omega_2$. A state machine $\tilde{P} = (\tilde{Q}, W, \tilde{\lambda}, \tilde{Q}_0) \subseteq P_l || P_{spec} = (Q, W, \lambda, Q_0)$ is a *controllable substructure* of $P_l || P_{spec}$ with respect to P_l if for every reachable state $(z, \chi) \in Z_l \times X_{spec}$ of $\tilde{P}$, a transition $((z, \chi), \omega, (z', \chi')) \in \lambda$ can only be an element in $\tilde{\lambda}$ if $((z, \chi), \omega', (z'', -)) \in \tilde{\lambda}$ for every partner (z, ω', z'') of (z, ω, z').

Let $\{P_{CN}\}$ be the set of all controllable substructures of $P_l || P_{spec}$ with respect to P_l that are both reachable and temporally nonblocking. $\{P_{CN}\}$ is closed under union (Moor et al., 2002). Hence, $\{P_{CN}\}$ forms a finite upper semi-lattice with the join operation $\cup$. Therefore, $P^+_{sup} :=$

$\sup\{P_{CN}\}$ uniquely exists. If $P^+_{sup} \neq \varnothing$, it serves as a realization of the least restrictive temporally nonblocking supervisor guaranteeing that the specifications are met for $\Sigma_l \cong P_l$ (and therefore Σ). If $P^+_{sup} = \varnothing$, no such supervisor can be computed for Σ on the basis of Σ_l.

We give an outline of the synthesis procedure in a variant that turns out useful for the refinement method proposed in the following section.

Algorithm 4. Least restrictive supervisor for P_l to enforce P_{spec}.
Set $\hat{P} := (\hat{Q}, W, \hat{\lambda}, \hat{Q}_0) := P_l || P_{spec}$.

1) Iteratively remove all blocking states and associated incoming transitions from $\hat{P}$.
2) Remove all transitions $((z,\chi),\omega_1,(z'_1,\chi')) \in \hat{\lambda}$ from $\hat{P}$ for which there exists a partner $(z,\omega_2,z'_2) \in \delta_l$ of (z,ω_1,z'_1) but $((z,\chi),\omega_2,-) \notin \hat{\lambda}$. If no transitions have been removed, go to Step 3; else, go to Step 1.
3) Set $\hat{Q}_{win} := \hat{Q}$ the current state set of $\hat{P}$.
4) Remove all unreachable states and associated outgoing transitions from $\hat{P}$.
5) Terminate the algorithm and set $P^+_{sup} = \hat{P}$.

The loop over Steps 1 and 2 removes states that block and transitions that violate the controllability condition. Hence, when entering Step 3, there are only states left that do not block and have a set of enabled events that can be enforced by control. If, for whatever reason, the plant abstraction P_l generates a finite sequence $\langle \omega_0, \omega_1, \ldots, \omega_k \rangle$ that drives $P_l||P_{spec}$ to a state within $\hat{Q}_{win}$, it can from then on be controlled to be temporally nonblocking and to satisfy the specification. We therefore refer to $\hat{Q}_{win}$ as the *winning states*. The set of all finite sequences that drive $P_l||P_{spec}$ to a winning state amounts to the *controllability prefix*, as introduced by Thistle & Wonham (1994b) for a more general class of ω-languages. In fact, when applying the algorithm presented in Thistle & Wonham (1994a) to the special case at hand, it simplifies to the above Steps 1–3. The post-processing Step 4 restricts the result to the reachable part. If and only if the initial state has not been removed and, hence, is a winning state, we end up with $P^+_{sup} \neq \varnothing$ and, hence, obtain a solution to the control problem. As a more common variant of the above algorithm, one could integrate the post-processing Step 4 into Step 1 to obtain the same final result P^+_{sup} and to gain some computational performance. However, in the case of $P^+_{sup} = \varnothing$, it is the intermediate result $\hat{Q}_{win}$ that will be useful in directing a refinement of the abstraction.

Example 5. For the water tank system in Example 2, assume the specification Σ_{spec} that after two time steps, the output E is not allowed to occur any more. A state machine P_{spec} realizing this specification is shown in Fig. 2.

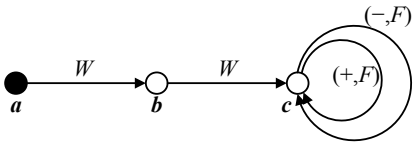

Fig. 2. Realization P_{spec} for Σ_{spec}.

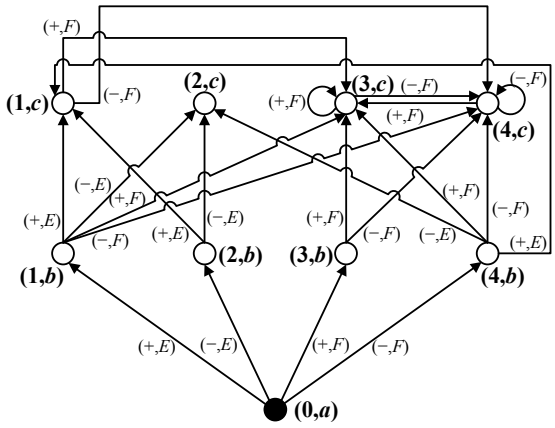

Fig. 3. Parallel composition $P_1||P_{spec}$.

Fig. 3 illustrates the parallel composition $P_1||P_{spec}$. Compared with Fig. 1, all the unacceptable trajectories are eliminated in the induced behavior. However, a closer examination of Figs 2 and 3 shows that derivation of P^+_{sup} from $P_1||P_{spec}$ results in $P^+_{sup} = \varnothing$. We go through the synthesis algorithm step by step. In $\hat{P} := P_1||P_{spec}$, state $(2,c)$ is blocking and, hence, is removed at Step 1 together with all incoming transitions. This introduces no additional blocking states and the algorithm proceeds with Step 2. Among the transitions from $(1,c)$, those associated with $(+,E)$ and $(-,E)$ are disabled in $\hat{P}$, and, hence, all their partners are disabled in Step 2: $(+,F)$ for $(+,E)$ and $(-,F)$ for $(-,E)$. Removing these transitions makes $(1,c)$ a blocking state. Likewise, $(4,c)$ is reduced to a blocking state by eliminating the respective partners. The algorithm now loops back to Step 1 and removes the blocking states $(1,c)$ and $(4,c)$ together with all incoming transitions. This turns $(2,b)$ also into a blocking state to be removed. The algorithm proceeds with Step 2, and, detects missing partners in states $(0,a)$, $(1,b)$ and $(4,b)$. This leads to the removal of the remaining outgoing transition of $(1,b)$ and$(4,b)$, respectively, and, hence, turns $(1,b)$ and $(4,b)$ into blocking states, to be removed in the next invocation of Step 1. In the subsequent iteration, $(0,a)$ becomes blocking for the same reason and also gets removed. The only remaining states $(3,b)$ and $(3,c)$ are non-blocking and do conform with the controllability condition. Hence neither the remaining states nor the remaining transitions are removed and the loop exits to Step 3 and sets $\hat{Q}_{win} = \{(3,b),(3,c)\}$. The positive reading here is that, if, for whatever reason, the first measurement turns out F then we can control the plant according to the specification by constantly applying the input symbol $+$. However, we also have that $(0,a) \notin \hat{Q}_{win}$ and, if the first measurement turns out E, the abstraction does not allow us to conclude that a controller can enforce the specification for all future. Consequently, the post-processing Step 4 yields $P^+_{sup} = \varnothing$.

As Example 5 demonstrates, if P^+_{sup} is found to be $\varnothing$, no supervisor can be synthesized on the basis of the strongest l-complete approximation Σ_l unless Σ_{spec} is somehow relaxed. If Σ_{spec} needs to be retained, (Moor & Raisch, 1999; Moor et al., 2002; Park & Raisch, 2015; Schmuck & Raisch, 2014) suggest to use a globally refined approximation, that is, Σ_{l+1} instead of Σ_l. But, as Theorem 1 hints, this global refinement corresponding to an increase in l

requires $\mathcal{B}|_{[0,l+1]}$ instead of $\mathcal{B}|_{[0,l]}$. In fact, it is easily derived that the state set Z_l of P_l has maximum cardinality $\sum_{k=1}^{l}(|U||Y|)^k$ excluding the initial state. Hence, as the approximation parameter l increases by one, the number of states in the realization P_l potentially increases by a multiplicative factor of $|U||Y|$.

4. LOCAL REFINEMENT FOR L-COMPLETE APPROXIMATIONS

As an alternative policy, we present a novel scheme of local refinement. The proposed strategy is applicable to the case where, given $P_l \cong \Sigma_l = (\mathbb{N}_0, W, \mathcal{B}_l)$ and $P_{spec} \cong \Sigma_{spec} = (\mathbb{N}_0, W, \mathcal{B}_{spec})$, $\mathcal{B}_l \cap \mathcal{B}_{spec} \neq \varnothing$ and therefore $P_l||P_{spec} \neq \varnothing$ but $P^+_{sup} = \varnothing$. Instead of deriving P_{l+1}, we refine $P_l||P_{spec}$ only at those states that violate the controllability condition. $\mathcal{B}_l \cap \mathcal{B}_{spec} \neq \varnothing$ means that the current approximation P_l is indeed compatible with some specified signal. However, on the basis of this approximation, the specification cannot be enforced in a nonblocking way by exclusively disabling input symbols. This is reflected in $P^+_{sup} = \varnothing$. In this case, we distinguish the following classes of states in order to construct a *local refinement*.

Definition 6. Given $P_l||P_{spec} = (Q, W, \lambda, Q_0)$, define the following.

(i) A *non-implementable state* is a state $q = (z, \chi) \in Q$ such that there exists a pair of partners $(z, \omega, z') \in \delta_l$ and $(z, \omega', z'') \in \delta_l$ with $\mathscr{P}_U\omega = \mathscr{P}_U\omega'$ in P_l for which $((z, \chi), \omega, (z', -)) \in \lambda$ but $((z, \chi), \omega', (z'', -)) \notin \lambda$. $Q_{ni} \subseteq Q$ denotes the set of non-implementable states.

(ii) $q = (z, \chi) \in Q$ is a *boundary state* if $z = \langle \omega_0, \ldots, \omega_{l-1} \rangle \in W^l$ and there exists $(z', \chi') \in Q$ such that $((z', \chi'), \omega_{l-1}, (z, \chi)) \in \lambda$ and $z' \in W^l$. $Q_{bd} \subseteq Q$ denotes the set of boundary states.

(iii) Execute the synthesis algorithm (Algorithm 4) on the input data P_l and P_{spec} and denote Q_{win} the *winning states*.

A non-implementable state of $P_l||P_{spec}$ is a state where the controllability condition is violated. A boundary state is a state for which the projection on the state set Z_l of P_l is a "boundary" one in the sense that it memorizes the maximum number l of external symbols and is reached from another state exhibiting the same property. A winning state is a state from which on liveness and safety can be enforced by a suitable controller.

Example 7. Consider $P_1||P_{spec}$ in Example 5, depicted in Fig. 3. As discussed in Example 5, the controllability condition is violated at states $(1, c)$ and $(4, c)$. Hence, $\{(1, c), (4, c)\} \subseteq Q_{ni}$. Moreover, as $((1, b), (+, E), (1, c)) \in \lambda$, $((3, b), (-, F), (4, c)) \in \lambda$, and $1, 3 \in W^1$, we have $\{(1, c), (4, c)\} \subseteq Q_{bd}$. By inspection, it is seen that $(1, c)$ and $(4, c)$ are the only non-implementable states. Therefore, $Q_{ni} = Q_{ni} \cap Q_{bd} = \{(1, c), (4, c)\}$. Recall from the previous section that $Q_{win} = \{(3, b), (3, c)\}$.

While the non-implementable states Q_{ni} are considered first candidates for a possible refinement, not all of them are expected to actually contribute to the solution of the synthesis problem at hand. To this end, we make three observations.

First, suppose $q \in Q_{ni} \cap Q_{bd}$. This implies the following.

(a) $q = (z, \chi)$ with $z = \langle \omega_1, \ldots, \omega_l \rangle$. Furthermore, there exists $q' = (z', \chi') \in Q$ with $z' = \langle \omega_0, \ldots, \omega_{l-1} \rangle$ such that $(q', \omega_l, q) \in \lambda$. Therefore, according to Theorem 1, $\langle \omega_0, \ldots, \omega_l \rangle \in \mathcal{B}|_{[0,l]}$.

(b) There exist $\omega, \omega' \in W$ with $\mathscr{P}_U\omega = \mathscr{P}_U\omega'$ such that $(z, \omega, \tilde{z}) \in \delta, (z, \omega', \tilde{z}') \in \delta$ for some $\tilde{z}, \tilde{z}' \in Z_l$ and $(q, \omega, \tilde{q}) \in \lambda, (q, \omega', -) \notin \lambda$ for some $\tilde{q} \in Q$.

Clearly, the violation of the controllability condition expressed by item (b) can potentially be resolved if we use a local approximation refinement in the sense of checking whether $\langle \omega_0, \ldots, \omega_l, \omega' \rangle \in \mathcal{B}|_{[0,l+1]}$. If the result of this check is negative, this string is not compatible with Σ_{l+1} and hence with Σ, and it will not occur in any closed-loop behavior involving the underlying hybrid plant Σ. In this case, there is no controllability issue for the locally refined approximation. We will refer to this procedure as a *local one-step refinement*. If the controllability issue in the state q can be resolved in this way, we will say that q is implementable with respect to P_l under local one-step refinement.

Second, suppose $q \in Q_{ni} \backslash Q_{bd}$. This implies the following.

(a) $q = (z, \chi)$ with $z = \langle \omega_0, \ldots, \omega_r \rangle$, $r \leqslant l - 1$. Furthermore, for all $q' = (z', \chi') \in Q$ such that $(q', \omega_r, q) \in \lambda$, we have $z' = \langle \omega_0, \ldots, \omega_{r-1} \rangle$. Therefore, according to Theorem 1, $\langle \omega_0, \ldots, \omega_r \rangle \in \mathcal{B}|_{[0,r]}$.

(b) same as item (b) above.

This implies that local refinement as introduced above would involve checking whether the condition

$$\langle \omega_0, \ldots, \omega_r, \omega' \rangle \in \mathcal{B}|_{[0,r+1]}$$

is valid. As $r + 1 \leqslant l$, this information is already encoded in the state machine P_l realizing Σ_l. Hence, local one-step refinement of states in $Q_{ni} \backslash Q_{bd}$ will not resolve controllability issues.

Our third and final observation is that once the system enters a state $q \in Q_{win}$, there exists a controller that faithfully operates the plant within Q_{win} and that this fact is already encoded in the abstraction P_l. In particular, there is no need for a refinement at states $Q_{win} \cap Q_{ni}$ since the synthesis procedure will succeed in resolving any implementation or blocking issues once Q_{win} is reached.

These considerations suggest the following procedure. Starting with $\hat{P} := P_l||P_{spec}$, we essentially apply the standard synthesis algorithm in that we iteratively remove blocking states and transitions that violate the controllability condition — except that we do not insist on controllability for transitions originating in the *refinement candidates* $\Xi := (Q_{ni} \cap Q_{bd}) \backslash Q_{win}$. For the latter states, we implement a local one-step refinement. This step either establishes implementability or fails to do so. In the former case, no further action is required for the respective state. In the latter case, the associated transitions are removed. This procedure is implemented in the following Algorithm 8. Its result, denoted by $\hat{P}^+_{sup}$, is the largest substructure of $P_l||P_{spec}$ that is reachable and temporally nonblocking and where all states are either implementable with respect to P_l or implementable with respect to P_l under local one-step refinement. It represents the least restrictive supervisor for P_l under local one-step refinement.

Algorithm 8. Least restrictive supervisor for P_l under local one-step refinement.
Given P_l and P_{spec} with $P_l||P_{spec} \neq \varnothing$, run Algorithm 4 to obtain the winning states $\hat{Q}_{win}$, then set $\hat{P} := (\hat{Q}, W, \hat{\lambda}, \hat{Q}_0) := P_l||P_{spec}$.

1) Iteratively remove all blocking or unreachable states and associated transitions from $\hat{P}$. Denote the resulting state machine again by $\hat{P}$ and identify the state sets $\hat{Q}_{ni} \subseteq \hat{Q}$ (the set of non-implementable states) and $\hat{Q}_{bd} \subseteq \hat{Q}$ (the set of boundary states) according to Definition 6. Set the refinement candidate states $\hat{\Xi} := (\hat{Q}_{ni} \cap \hat{Q}_{bd}) \backslash \hat{Q}_{win}$.
2) If $\hat{Q}_{ni} \backslash \hat{\Xi} = \varnothing$, proceed to Step 3. Else, remove all transitions originating in $\hat{Q}_{ni} \backslash \hat{\Xi}$ that violate the controllability condition according to Definition 3 and return to Step 1.
3) If $\hat{\Xi} = \varnothing$, go to Step 11. Else, let $\hat{\Xi} := \{q_1, \ldots, q_n\}$ and set $i := 1$ and $\mathcal{C} := 0$.
4) For $q_i = (z_i, \chi_i) \in \hat{\Xi}$, let $z_i := \langle \omega_0, \ldots, \omega_{l-1} \rangle$ and let $(z_i, \omega, z'), (z_i, \omega', z'') \in \delta_l$ be a pair of partners for which $((z_i, \chi_i), \omega, (z', -)) \in \hat{\lambda}$ and $((z_i, \chi_i), \omega', (z'', -)) \notin \hat{\lambda}$. Denote the set of all such ω''s by $\Psi(z_i) := \{\omega'_1, \ldots \omega'_p\} \subseteq W$.
5) Let $Q_{(z_i,\chi_i)} := \{(z, \chi) \in \hat{Q} | ((z, \chi), \omega_{l-1}, (z_i, \chi_i)) \in \hat{\lambda},\ z \in Z_l \cap W^l\}$, and let
$$\begin{aligned} \mathscr{P}_{Z_l} Q_{(z_i,\chi_i)} &:= \{z | \exists \chi \text{ s.t. } (z, \chi) \in Q_{(z_i,\chi_i)}\} \\ &:= \{\zeta_1, \ldots, \zeta_m\}. \end{aligned}$$
Set $j := 1$.
6) If, for any $\omega'_k \in \Psi(z_i)$, $\langle \zeta_j, \omega_{l-1}, \omega'_k \rangle \in \mathcal{B}|_{[0,l+1]}$, remove the transition $((\zeta_j, -), \omega_{l-1}, (z_i, \chi_i)) \in \hat{\lambda}$ from $\hat{P}$ and set $\mathcal{C} := 1$.
7) If $j < m$, set $j := j + 1$ and return to Step 6.
8) If there exists a transition $((\zeta, -), \omega_{l-1}, (z_i, \chi_i)) \in \hat{\lambda}$ with $\zeta \in W^{l-1}$, remove this transition from $\hat{P}$ and set $\mathcal{C} := 1$.
9) If $i < n$, set $i := i + 1$ and return to Step 4.
10) If $\mathcal{C} = 1$, return to Step 1.
11) Terminate the algorithm and set $\hat{P}^+_{sup} = \hat{P}$. If $\hat{P}^+_{sup} \neq \varnothing$, it is the least restrictive supervisor for P_l under local one-step refinement.

This algorithm is based on the procedure for deriving the least restrictive supervisor for P_l described in Moor et al. (2002). It additionally incorporates local one-step refinement in Steps 4–9. In Step 4, for all refinement candidates $q_i = (z_i, \chi_i)$, we collect in $\Psi(z_i) \subseteq W$ all external symbols that have been disabled in z_i by forming the composition $P_l||P_{spec}$ without respecting the controllability requirements. In Step 6, we check for every $\omega' \in \Psi(z_i)$ whether the string $\langle \zeta_j, \omega_{l-1}, \omega' \rangle \in W^{l+2}$ belongs to $\mathcal{B}|_{[0,l+1]}$. If this is true for at least one $\omega' \in \Psi(z_i)$, it implies that the controllability issue is also present under local one-step refinement. Therefore, we need to erase $((\zeta_j, -), \omega_{l-1}, (z_i, \chi_i))$ from $\hat{P}$. On the other hand, $\langle \zeta_j, \omega_{l-1}, \omega' \rangle \notin \mathcal{B}|_{[0,l+1]}$ for all $\omega' \in \Psi(z_i)$ implies that the controllability problem observed for P_l will not be an issue for P_{l+1} and hence for the underlying hybrid system Σ and its realization P. We therefore keep the transition $((\zeta_j, -), \omega_{l-1}, (z_i, \chi_i))$ in $\hat{P}$. In step 8, we check whether there are transitions $((\zeta, -), \omega_{l-1}, (z_i, \chi_i)) \in \hat{\lambda}$ with $\zeta \in W^{l-1}$. This is only possible if $\zeta = \langle \omega_0, \ldots, \omega_{l-2} \rangle$ (for $l \geqslant 2$) or $\zeta = \omega^*$ (if $l = 1$). Such transitions always need to be removed since $\langle \zeta, \omega_{l-1}, \omega'_k \rangle \in \mathcal{B}|_{[0,l]}$.

Note that after completing the procedure of local one-step refinement, we return to Step 1 if the indicator parameter $\mathcal{C}$ is 1 (Step 10). This is because the removal of a transition leading to a boundary non-implementable state in Step 6 may render elements in the state set of the updated $\hat{P}$ blocking, unreachable, or non-implementable.

Let us continue to denote by $\hat{Q}_{ni}$ and $\hat{\Xi}$ the non-implementable state set and the refinement candidate set of $\hat{P}^+_{sup}$, respectively. If $\hat{P}^+_{sup} \neq \varnothing$, clearly $\hat{Q}_{ni} \backslash \hat{\Xi} = \varnothing$. If $\hat{\Xi} \neq \varnothing$, i.e., there are states that are non-implementable for P_l, they are implementable for P_l under local one-step refinement. To prove that a nonempty $\hat{P}^+_{sup}$ generated by Algorithm 8 is a valid supervisor for the underlying hybrid system Σ, we utilize the known fact that P is not restrained by composition with P_l.

Proposition 9. (Moor et al., 2002) Let $(x, z) \in X \times Z_l$ be a reachable state of $P||P_l$ and assume $(x, \omega, -) \in \delta$ for some $\omega \in W$. Then $(z, \omega, -) \in \delta_l$.

Theorem 10. Assume that $\hat{P}^+_{sup} \neq \varnothing$ is derived by applying Algorithm 8. Then the closed-loop realization $P||\hat{P}^+_{sup}$ with its next state relation λ_{cl} is temporally nonblocking and is a controllable substructure with respect to P.

Proof: Let $P = (X, W, \delta, X_0)$, $P_l = (Z_l, W, \delta_l, Z_0)$, $P_{spec} = (X_{spec}, W, \delta_{spec}, X_{spec0})$, $P_l||P_{spec} = (Q, W, \lambda, Q_0)$, and $\hat{P}^+_{sup} = (\hat{Q}, W, \hat{\lambda}, Q_0)$. Also, let $\mathcal{B}_s$ be the full behavior induced by P, and let $\hat{\mathcal{B}}_{sup}$ be the external behavior induced by $\hat{P}^+_{sup}$.

(i) Temporally nonblocking: Let (x, z, χ) be a reachable state of $P||\hat{P}^+_{sup}$. Then, (x, z) and (z, χ) are reachable states of $P||P_l$ and $\hat{P}^+_{sup}$, respectively. Since $\hat{P}^+_{sup}$ is temporally nonblocking by definition, there exist $\omega \in W$ and $z' \in Z_l$ such that $((z, \chi), \omega, (z', -)) \in \hat{\lambda}$. Also, as P is an I/S/O machine, there exists $\omega' \in W$ with $\mathscr{P}_U \omega' = \mathscr{P}_U \omega$ such that $(x, \omega', x') \in \delta$ for some $x' \in X$. Since (x, z) is a reachable state of $P||P_l$ and $(x, \omega', x') \in \delta$, $(z, \omega', z'') \in \delta_l$ for some $z'' \in Z_l$ by Proposition 9. Now consider (z, χ). If $(z, \chi) \in \hat{Q} \backslash \hat{\Xi}$, Algorithm 8 in Steps 1 and 2 establishes implementability. Thus $((z, \chi), \omega, (z', -)) \in \hat{\lambda}$ implies $((z, \chi), \omega', (z'', -)) \in \hat{\lambda}$, which leads to $((x, z, \chi), \omega', (x', z'', -)) \in \lambda_{cl}$. Else if $(z, \chi) \in \hat{\Xi}$, it is a boundary non-implementable state with, say, $z = \langle \omega_0, \ldots, \omega_{l-1} \rangle$. We claim that $((z, \chi), \omega', (z'', -)) \in \hat{\lambda}$ still holds in this case. To show this, consider ω'' such that $\mathscr{P}_U \omega'' = \mathscr{P}_U \omega' = \mathscr{P}_U \omega$ and $((z, \chi), \omega'', (-, -)) \notin \hat{\lambda}$. This implies, by the construction rules of $\hat{P}^+_{sup}$, that for all incoming transitions $((\zeta_j, -), \omega_{l-1}, (z, \chi)) \in \hat{\lambda}$, we have $\zeta_j \in Z_l \cap W^l$ and $\langle \zeta_j, \omega_{l-1}, \omega'' \rangle \notin \mathcal{B}|_{[0,l+1]}$. As the state (z, χ) in $\hat{P}^+_{sup}$ can only be reached via strings $\langle \zeta_j, \omega_{l-1} \rangle$ with $\zeta_j \in Z_l \cap W^l$ and as (x, z, χ) is a reachable state of $P||\hat{P}^+_{sup}$, the state x of P is also reachable via some $\langle \zeta_j, \omega_{l-1} \rangle$. Consequently, $(x, \omega'', -) \notin \delta$ (otherwise

$\langle \zeta_j, \omega_{l-1}, \omega'' \rangle \in \mathcal{B}|_{[0,l+1]}$), and hence $\omega'' \neq \omega'$. Therefore $((z,\chi),\omega',(z'',-)) \in \hat{\lambda}$ and $((x,z,\chi),\omega',(x',z'',-)) \in \lambda_{cl}$.

(ii) Controllable substructure: Let (x,z,χ) be a reachable state of $P||\hat{P}^+_{sup}$. Let $\omega \in W$ and $(x',z',\chi') \in X \times Z_l \times X_{spec}$ such that $((x,z,\chi),\omega,(x',z',\chi')) \in \lambda_{cl}$. Let $(x,\omega',x'') \in \delta$ be a partner of $(x,\omega,x') \in \delta$, i.e., $\mathscr{P}_U\omega' = \mathscr{P}_U\omega$. Since $((x,z,\chi),\omega,(x',z',\chi')) \in \lambda_{cl}$, $((z,\chi),\omega,(z',\chi')) \in \hat{\lambda}$. By Proposition 9, $(z,\omega,z') \in \delta_l$ and $(z,\omega',z'') \in \delta_l$ for some $z'' \in Z_l$. First, assume that $(z,\chi) \in \hat{Q}\backslash\hat{\Xi}$. Again by Steps 1 and 2 of Algorithm 8, the implementability issue for (z,χ) is resolved. Hence $((z,\chi),\omega',(z'',-)) \in \hat{\lambda}$ and $((x,z,\chi),\omega',(x'',z'',-)) \in \hat{\lambda}_{cl}$. Next, assume that $(z,\chi) \in \hat{\Xi}$. Recall from the foregoing discussion that $((z,\chi),\omega',(z'',-)) \in \hat{\lambda}$ still holds true. Therefore, we have $((x,z,\chi),\omega',(x'',z'',-)) \in \hat{\lambda}_{cl}$. □

Example 11. Let us apply Algorithm 8 to $P_1||P_{spec}$ in Example 5. Referring to Fig. 3, we first remove the blocking state $(2,c)$ and the associated transitions (Step 1). In the resulting state machine $\hat{P}$, states $(1,c)$ and $(4,c)$ are in $\hat{\Xi}$, and $(1,b),(4,b) \in \hat{Q}_{ni}\backslash\hat{\Xi}$. Hence, we have to remove transitions originating in the latter states that violate the conditions of controllability, i.e., $((1,b),(-,F),(4,c))$ and $((4,b),(-,F),(4,c))$ (Step 2). This results in the state machine $\hat{P}$ shown in Fig. 4. As $\hat{\Xi} = \{(1,c),(4,c)\} := \{q_1,q_2\} \neq \varnothing$, we set $\mathcal{C} = 0$ and proceed to local refinement for $i = 1,2$ in Steps 4 to 9.

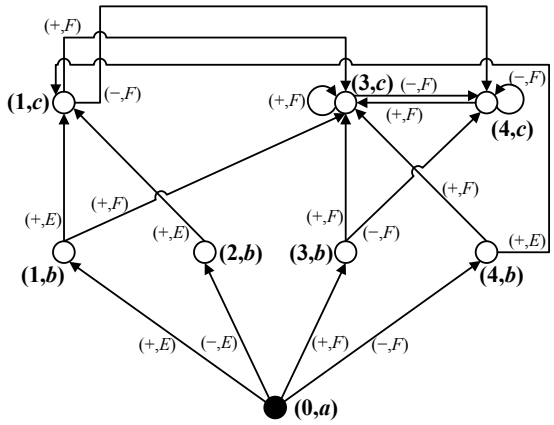

Fig. 4. $\hat{P}$ obtained by applying Steps 1 and 2 of Algorithm 8 to $P_1||P_{spec}$ of Fig. 3.

$i = 1$, $q_1 = (1,c)$. Comparing Figs. 1 and 4, we obtain $\Psi(1) = \{(+,E),(-,E)\}$, where Ψ is defined in Step 4 of Algorithm 8. Referring to Fig. 4, we have

$$Q_{(1,c)} = \{(1,b),(2,b),(4,b)\},$$
$$\mathscr{P}_{Z_l} Q_{(1,c)} = \{1,2,4\} = \{(+,E),(-,E),(-,F)\} := \{\zeta_1,\zeta_2,\zeta_3\},$$

with both sets as defined in Step 5 of Algorithm 8. In Step 6, for

$j = 1$, we check whether $\langle (+,E),(+,E),(+,E)\rangle$ or $\langle (+,E),(+,E),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. From (3) describing the input/output behavior of the hybrid system Σ in Example 2, we can readily deduce that this is not the case. Hence, we proceed to

$j = 2$, where we check whether $\langle (-,E),(+,E),(+,E)\rangle$ or $\langle (-,E),(+,E),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. This is indeed true for both strings (as seen from (3)). Hence, the transition $((2,b),(+,E),(1,c))$ is removed from $\hat{P}$, and we set $\mathcal{C} = 1$. We proceed to

$j = 3$, where we check whether $\langle (-,F),(+,E),(+,E)\rangle$ or $\langle (-,F),(+,E),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. (3) reveals that this is not the case. Hence, no further transitions need to be removed.

As the "if" condition in Step 8 is not true, no action is required.

$i = 2$, $q_2 = (4,c)$. Comparing Figs. 1 and 4, we obtain $\Psi(4) = \{(+,E),(-,E)\}$. From Fig. 4, we see that

$$Q_{(4,c)} = \{(3,b),(1,c),(3,c),(4,c)\},$$
$$\mathscr{P}_{Z_l} Q_{(4,c)} = \{1,3,4\} = \{(+,E),(+,F),(-,F)\} := \{\zeta_1,\zeta_2,\zeta_3\}.$$

$j = 1$. Check whether $\langle (+,E),(-,F),(+,E)\rangle$ or $\langle (+,E),(-,F),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. This is true, hence the transition $((1,c),(-,F),(4,c))$ is removed.

$j = 2$. Check whether $\langle (+,F),(-,F),(+,E)\rangle$ or $\langle (+,F),(-,F),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. This is not the case. Hence, no action is required.

$j = 3$. Check whether $\langle (-,F),(-,F),(+,E)\rangle$ or $\langle (-,F),(-,F),(-,E)\rangle$ is a string in $\mathcal{B}|_{[0,2]}$. This is true for both strings. Hence, the transition $((4,c),(-,F),(4,c))$ is removed.

As the "if" condition in Step 8 is not true, no action is required.

As $\mathcal{C} = 1$ (Step 10), we return to Step 1 of Algorithm 8, remove all the blocking and unreachable states from the truncated finite state machine $\hat{P}$, and iterate from Step 2. Fig. 5 shows the final result of Algorithm 8, $\hat{P}^+_{sup}$. It represents the least restrictive supervisor for the

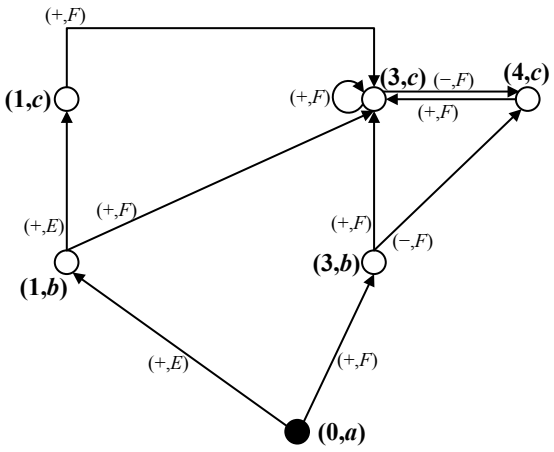

Fig. 5. $\hat{P}^+_{sup}$ for $P_1||P_{spec}$ of Fig. 3.

strongest 1− complete approximation of Σ under local one-step refinement. According to Theorem 10, it is also a valid supervisor for the the underlying hybrid system Σ, respectively its realization P.

The above example demonstrates the contribution of this paper. Namely, if strongest l-complete approximation $\Sigma_l \cong$

P_l of a given hybrid system $\Sigma \cong P$ does not allow computation of a supervisor enforcing a given specification $\Sigma_{spec} \cong P_{spec}$ for Σ, it may be possible, by applying Algorithm 8, to compute such a supervisor by locally refining $P_l||P_{spec}$ in specific states instead of determining a globally refined approximation, $\Sigma_{l+1} \cong P_{l+1}$.

5. CONCLUSION

We have addressed a question arising in approximation-based control of hybrid systems. More specifically, the l-complete approximation scheme (Moor & Raisch, 1999; Moor et al., 2002) provides the possibility of generating a sequence of finite state abstractions P_l, $l = 1, 2, \ldots$ for a (possibly infinite state) system with discrete-valued external signals. Both approximation accuracy and complexity are monotonous in the parameter l, with the cardinality of the abstraction state set growing exponentially in l. Given a finite-state specification P_{spec}, controller synthesis within this scheme typically proceeds in an iterative fashion. Start with $l = 1$, i.e., the coarsest abstraction, form the composition $P_l||P_{spec}$, and search for the largest controllable and temporally nonblocking substructure P^+_{sup} of $P_l||P_{spec}$. If P^+_{sup} is non-empty, it represents the least restrictive supervisor for P_l that enforces P_{spec}. It is also a valid, though not necessarily the least restrictive, supervisor for the underlying hybrid system P, i.e., P under control of P^+_{sup} will be temporally nonblocking, respect the controllability condition and satisfy the specification (Moor & Raisch, 1999; Moor et al., 2002). If, however, $P^+_{sup} = \varnothing$, supervisor synthesis on the basis of P_l has failed, and a globally refined abstraction, P_{l+1}, is generated, followed by supervisor synthesis for P_{l+1}. As the cardinality of the approximation state set is exponential in l, this may be computationally prohibitive. Motivated by this fact, and inspired by Moor et al. (2006), we have presented a local refinement scheme. More precisely, refinement is only undertaken in certain states of $P_l||P_{spec}$ violating the controllability condition. If this procedure, summarized in Algorithm 8, terminates with a nonempty state machine $\hat{P}^+_{sup}$, the latter can be interpreted as the least restrictive supervisor for P_l under local one-step refinement. As shown in Theorem 10, it is also a valid supervisor for the underlying hybrid system P. The execution of the proposed algorithm has been demonstrated for a simple example.

REFERENCES

Alur, R., Henzinger, T., Lafferriere, G., & Pappas, G. (2000). Discrete abstractions of hybrid systems. *Proceedings of the IEEE, 88*(7), 971–984.

Cassandras, C. G., & Lafortune, S. (2008). *Introduction to discrete event systems (2nd ed.).* New York, NY: Springer-Verlag.

Clarke, E., Fehnker, A., Han, Z., Krogh, B., Ouaknine, J., Stursberg, O., & Theobald, M. (2003). Abstraction and counterexample-guided refinement in model checking of hybrid systems. *International Journal of Foundations of Computer Science, 14*(4), 583–604, 2003.

Girard, A., & Pappas, G. J. (2009). Hierarchical control system design using approximate simulation. *Automatica, 45*(2), 566–571.

Moor, T., Davoren, J. M., & Raisch, J. (2006). Learning by doing: systematic abstraction refinement for hybrid control synthesis. *IEE Proceedings: Control Theory and Applications, 153*(5), 591–599.

Moor, T., & Raisch, J. (1999). Supervisory control of hybrid systems within a behavioural framework. *Systems & Control Letters, 38*(3), 157–166.

Moor, T., Raisch, J., & O'Young, S. (2002). Discrete supervisory control of hybrid systems based on l-complete approximations. *Discrete Event Dynamic Systems: Theory and Applications, 12*, 83–107.

Park, S. J., & Raisch, J. (2015). Supervisory control of hybrid systems under partial observation based on l-complete approximations. *IEEE Transactions on Automatic Control, 60*(5), 1404–1409.

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

Ramadge, P. J., & Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of the IEEE, 77*(1), 81–98.

Schmuck, A. K., & Raisch, J. (2014). Asynchronous l-complete approximations. *Systems & Control Letters, 73*, 67–75.

Stursberg, O. (2006). Supervisory control of hybrid systems based on model abstraction and guided search. *Nonlinear Analysis: Theory, Methods & Applications, 65*(6), 1168–1187.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*, New York, NY: Springer-Verlag.

Tarraf, D. C. (2014). An input-output construction of finite state ρ/μ approximations for control design. *IEEE Transactions on Automatic Control, 59*(12), 3164–3177.

Thistle, J. G., & Wonham, W. M. (1994a). Control of infinite behavior of finite automata, *SIAM Journal on Control and Optimization, 32*(4), 1075–1097.

Thistle, J. G., & Wonham, W. M. (1994b). Supervision of infinite behavior of discrete event systems. *SIAM Journal on Control and Optimization, 32*(4), 1098–1113.

Willems, J. C. (1991). Paradigms and puzzles in the theory of dynamic systems. *IEEE Transactions on Automatic Control, 36*(3), 258–294.