Strategic Refinements in Abstraction Based Supervisory Control of Hybrid Systems

Thomas Moor, J.M. Davoren
Research School of
Information Sciences and Engineering
Australian National University
Canberra ACT 0200, Australia
{*thomas.moor, j.m.davoren*}*@anu.edu.au*

Jörg Raisch
Otto-von-Guericke Universität, and
Max-Planck-Institut für
Dynamik komplexer technischer Systeme
D-39120 Magdeburg, Germany
raisch@mpi-magdeburg.mpg.de

Abstract

A common approach to hybrid control problems is to alternate refinement of a plant abstraction with trial controller synthesis performed on the current abstraction. These steps are repeated until a solution to the control problem can be found, or computational resources are exhausted. In this contribution we use a temporal decomposition of the control problem in order to gain relevant diagnostic information for those steps when synthesis fails. We use this information to focus the abstraction refinement on those features of the plant that are most crucial for the synthesis task at hand. This work is an advance over earlier abstraction based synthesis procedures which use an unfocused and global refinement on the plant model.

1. Introduction

In the basic hybrid control configuration, a switched plant with continuous dynamics accepts discrete inputs which select which of a finite number of vector fields is to be active. The plant exhibits discrete outputs via an A/D map which converts continuous states (or outputs) into discrete symbols. A widely accepted detailed model of this hybrid control configuration are so called *hybrid automata*; e.g. [1, 3]. In this paper, we take the perspective of a discrete supervisor that exclusively interacts with the hybrid plant via discrete external events. Here, an adequate model of the hybrid plant is the *external behaviour*, defined as the set of all sequences of pairs of input and output symbols that are compatible with the plant dynamics. Control specifications are formalised as languages over the same alphabet of pairs of input and output symbols, and the task of the supervisor is to restrict the plant behaviour so that the closed loop is guaranteed to only evolve on acceptable trajectories within the specification. A crucial feature of the hybrid setting is that state machine realisations of the plant typically evolve on a real-vector valued, and hence uncountable, state space. The core idea of abstraction based approaches is that, rather than synthesising a supervisor for the actual plant behaviour, one works instead with a plant abstraction that can be realised by a finite automaton; e.g. [4, 5, 6, 9]. In [7, 8], we develop *l-complete abstractions* to allow for the *refinement* of abstractions in the case that controller synthesis is not successful because of the abstraction being too coarse. The high-level strategy is to alternate trial controller synthesis with refinement of the abstraction, until either synthesis succeeds or computational resources are exhausted.

In the present paper, we undertake a more general study of the role and strategic use of refinements of abstractions for the purpose of supervisory controller synthesis. Two basic questions motivate our present study. How are we to formulate abstractions so that they readily allow for refinements? How are we to link the alternation of the synthesis and the abstraction refinement procedures so that a supervisor produced by the process is guaranteed to solve the original control problem for the hybrid plant? In addressing these questions, we develop a new approach to adaptively linking the synthesis and refinement steps. Our synthesis procedure not only reports failure, but in a precise sense locates the potential reason in the current abstraction. The refinement procedure then focuses its efforts on those aspects of the abstraction that have "caused" the failure in synthesis, rather than doing an unfocused global refinement.

The paper is organised as follows. Section 2 summarises key results from our earlier work in abstraction based supervisory controller synthesis for hybrid systems. In Section 3, we introduce the notion of a finite experiment to provide a flexible tool for model abstraction. Section 4 investigates the decomposition of a control problem in a start up phase and a long term component. In Section 5, we use the temporal decomposition to extract diagnostic information on an unsuccessful controller synthesis and we use this information to focus the efforts in the abstraction refinement.

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 IEEE

2. Abstraction based supervisory control

The purpose of this section is to summarise key results of our earlier work in abstraction based supervisory controller synthesis for hybrid systems [7, 8]. In the cited papers, we discuss both abstraction and supervisory controller synthesis within J.C. Willems' behavioural system theory; e.g. [11]. The choice of a common framework for both tasks facilitates a consistent discussion of the important question whether a supervisor that enforces a specification for a plant abstraction will also solve the problem for the original plant.

In [11], the behaviour of a dynamical system is introduced as the set of all trajectories on which the system can possibly evolve. Thus, behaviours represent dynamics in a similar way as *formal languages* are used to model DESs.

Definition 2.1. (See [11]) A *behaviour* $\mathfrak{B}$ *over* W is a subset $\mathfrak{B} \subseteq W^{\mathbb{N}_0} := \{w : \mathbb{N}_0 \to W\}$. [1] □

Our target class of hybrid plants inherits its input/output structure from the underlying continuous dynamics. Consequently, we assume throughout this paper that W the product composition of an input- and an output-component; i.e. $W = U \times Y$. This contrasts with the common practice in DES theory of working with the disjoint union of controllable and uncontrollable events. Willems characterises the traditional notion of inputs and outputs as follows.

Definition 2.2. (See [11], also [7]) A behaviour $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ is said to be an *I/- behaviour* if [2]

(i) the *input is free*, i.e. for all $u \in U^{\mathbb{N}_0}$ exists a $y \in Y^{\mathbb{N}_0}$ such that $(u, y) \in \mathfrak{B}$; and

(ii) the *output does not anticipate the input*, i.e. for all $k \in \mathbb{N}_0$, $(\tilde{u}, \tilde{y})$, $(\hat{u}, \hat{y}) \in \mathfrak{B}$, $\tilde{u}|_{[0,k)} = \hat{u}|_{[0,k)}$ there exists $(u, y) \in \mathfrak{B}$ such that $y|_{[0,k)} = \tilde{y}|_{[0,k)}$ and $u = \hat{u}$. □

Following the concepts of P.J. Ramadge and W.M. Wonham's supervisory control theory for DESs [10], the task of a supervisor $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}_0}$ is to restrict a plant behaviour $\mathfrak{B}_{\text{p}} \subseteq W^{\mathbb{N}_0}$ so that the closed loop $\mathfrak{B}_{\text{cl}} := \mathfrak{B}_{\text{p}} \cap \mathfrak{B}_{\text{sup}}$ is guaranteed to only evolve within the set of acceptable signals $\mathfrak{B}_{\text{spec}} \subseteq W^{\mathbb{N}_0}$; i.e. we aim for $\mathfrak{B}_{\text{cl}} \subseteq \mathfrak{B}_{\text{spec}}$. However, when interconnecting the plant and the supervisor one needs to ensure that: (i) the supervisor may enable or disable certain input events at any time but no restrictions must be imposed on the plant outputs; (ii) at any time there is a possible future evolution, so the closed-loop must not "get stuck". We state our *admissibility* conditions in terms of behaviours:

Definition 2.3. (See [7] [3]) Let $\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}_0}$. Then

(i) $\mathfrak{B}_{\text{sup}}$ is *generically implementable* if $k \in \mathbb{N}_0$, $(u, y)|_{[0,k]} \in \mathfrak{B}_{\text{sup}}|_{[0,k]}$, $(\tilde{u}, \tilde{y})|_{[0,k]} \in W^{k+1}$, $\tilde{u}|_{[0,k]} = u|_{[0,k]}$, $\tilde{y}|_{[0,k)} = y|_{[0,k]}$ implies that $(\tilde{u}, \tilde{y})|_{[0,k]} \in \mathfrak{B}_{\text{sup}}|_{[0,k]}$.

(ii) $\mathfrak{B}_{\text{p}}$ and $\mathfrak{B}_{\text{sup}}$ are *non-conflicting* if $\mathfrak{B}_{\text{p}}|_{[0,k]} \cap \mathfrak{B}_{\text{sup}}|_{[0,k]} = (\mathfrak{B}_{\text{p}} \cap \mathfrak{B}_{\text{sup}})|_{[0,k]}$ for all $k \in \mathbb{N}_0$. □

These definitions lead to the following formulation of the problem of supervisory control.

Definition 2.4. (See [7]) Given a plant $\mathfrak{B}_{\text{p}} \subseteq W^{\mathbb{N}_0}$ and a specification $\mathfrak{B}_{\text{spec}} \subseteq W^{\mathbb{N}_0}$, the pair $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$ is a *supervisory control problem*. A supervisor $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}_0}$ is *admissible* to the plant $\mathfrak{B}_{\text{p}}$, if $\mathfrak{B}_{\text{p}}$ and $\mathfrak{B}_{\text{sup}}$ are non-conflicting and $\mathfrak{B}_{\text{sup}}$ is generically implementable. A supervisor $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}_0}$ *enforces the specification* $\mathfrak{B}_{\text{spec}} \subseteq W^{\mathbb{N}_0}$ if $\mathfrak{B}_{\text{cl}} := \mathfrak{B}_{\text{p}} \cap \mathfrak{B}_{\text{sup}} \subseteq \mathfrak{B}_{\text{spec}}$. A supervisor $\mathfrak{B}_{\text{sup}}$ that is admissible to $\mathfrak{B}_{\text{p}}$ and that enforces $\mathfrak{B}_{\text{spec}}$ is said to be a *solution* of $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$. A solution $\mathfrak{B}_{\text{sup}}$ is *nontrivial* if it imposes a nontrivial closed loop behaviour $\mathfrak{B}_{\text{cl}} = \mathfrak{B}_{\text{p}} \cap \mathfrak{B}_{\text{sup}} \neq \emptyset$. □

It can be seen that a solution $\mathfrak{B}_{\text{sup}}$ is trivial if and only if $\mathfrak{B}_{\text{sup}} = \emptyset$, regardless $\mathfrak{B}_{\text{p}}$ and $\mathfrak{B}_{\text{spec}}$. In the spirit of [10], we obtain the unique existence of a least restrictive solution:

Corollary 2.5. (See [7]) Let $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$ denote a supervisory control problem. The set of all solutions of $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$ is a complete upper semi-lattice with the usual settheoretic operators "$\cup$" and the partial order "$\subseteq$". The supremal element $\mathfrak{B}_{\text{sup}}^{\downarrow}$ of that lattice is referred to as *least restrictive solution* of $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$. □

If both $\mathfrak{B}_{\text{p}}$ and $\mathfrak{B}_{\text{sup}}$ were realised by finite automata, the supervisory control problem could be readily solved with a slight modification of DES tools. However, since hybrid plants $\mathfrak{B}_{\text{p}}$ almost never have a finite realisation, we instead work with an *abstraction* $\mathfrak{B}_{\text{ca}}$, $\mathfrak{B}_{\text{p}} \subseteq \mathfrak{B}_{\text{ca}}$, that is realised by a finite automaton. For *complete* I/- behaviours $\mathfrak{B}_{\text{p}}$, Theorem 2.6 guarantees that any solution of $(\mathfrak{B}_{\text{ca}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$ also solves the original problem $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$. [4]

Theorem 2.6. (See [7]) Let $\mathfrak{B}_{\text{ca}} \subseteq W^{\mathbb{N}_0}$ be an abstraction of an I/- behaviour $\mathfrak{B}_{\text{p}} \subseteq W^{\mathbb{N}_0}$, let $\mathfrak{B}_{\text{spec}} \subseteq W^{\mathbb{N}_0}$, and let $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}_0}$ be a nontrivial solution to $(\mathfrak{B}_{\text{ca}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$. If $\mathfrak{B}_{\text{p}}$ and $\mathfrak{B}_{\text{sup}}$ are complete then $\mathfrak{B}_{\text{sup}}$ is a nontrivial solution of $(\mathfrak{B}_{\text{p}}, \mathfrak{B}_{\text{spec}})_{\text{svc}}$. □

Theorem 2.6 contrasts with the basic DES setting: suppose a supervisor has been synthesised in the framework provided by e.g. [10] and suppose that synthesis has been

[1] $\mathbb{N}$ denotes the positive integers, $\mathbb{N}_0$ denotes the nonnegative integers.

[2] The restriction operator $(\cdot)|_{[k_1,k_2)}$ maps sequences $w \in W^{\mathbb{N}_0}$ to finite strings $w|_{[k_1,k_2)} := w(k_1)w(k_1+1)\cdots w(k_2-1) \in W^{k_2-k_1}$ where $k_1, k_2 \in \mathbb{N}_0$, $k_1 \leq k_2$, and $W^0 := \{\epsilon\}$. Let $(\cdot)|_{[k_1,k_2]} := (\cdot)|_{[k_1,k_2+1)}$.

[3] The definition here of *generic implementability* corresponds to *implementability w.r.t. a particular plant* in [7], and it can be shown that the alternative formulation leads to precisely the same closed-loop behaviours.

[4] A behaviour $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ is said to be *complete* if [$w \in \mathfrak{B} \Leftrightarrow \forall\, k \in \mathbb{N}_0 : w|_{[0,k)} \in \mathfrak{B}|_{[0,k)}$]; see [11]. Regarding Theorem 2.6, the completeness condition on $\mathfrak{B}_{\text{p}}$ is not necessary, and alternative results exist for behaviours that are realised by so called *I/S/- state machines*; e.g. [8].

based on a plant abstraction that in some states can take transitions that the original plant cannot take; such a supervisor may rely on these "artifact transitions" and disable all other events in certain states; hence, the synchronous product of the supervisor with the original plant may block, and the corresponding languages fail to be nonconflicting.

3. Experiments

Were it possible to observe a phenomenon for infinite time, the behaviour describes all *possible* outcomes. In practice, one finds oneself restricted to finite time, and we formally define the outcome of such an *experiment* as a set of strings that are bounded in length. In this section, we develop an abstract notion of experiments to provide a general tool for generating models from other models.

Definition 3.1. A set of finite strings $S \subseteq W^* := \cup_{l\in\mathbb{N}_0} W^l$ is an *experiment over* W if there exists a $k \in \mathbb{N}_0$ such that $|s| \leq k$ for all $s \in S$. [5] [6] □

For practical purposes, we restrict experiments to be finite sets, i.e. $|S| \in \mathbb{N}_0$. [7] For the following discussion, the weaker assumption of a bounded length is sufficient. When the signal space W is a finite set, both conditions are equivalent. Rather than perform an experiment on the actual phenomenon, our interest here is on constructing models from other models, and therefore we introduce the notion of an experiment *on* a behaviour.

Definition 3.2. Let $S \subseteq W^*$ and $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ be an experiment and a behaviour, respectively. Then S *is an experiment on* $\mathfrak{B}$ if the following conditions are fulfilled:

(i) $s \in S \;\Rightarrow\; s \in \mathfrak{B}|_{[0,|s|)}$,

(ii) $w \in \mathfrak{B} \;\Rightarrow\; \exists\, l \in \mathbb{N}_0 : w|_{[0,l)} \in S$. □

By the first condition, an experiment must not hold strings that cannot occur according to the behaviour. The second condition requires an experiment to give some account of each trajectory that can occur.

Suppose we know S and we know that S is an experiment on some behaviour $\mathfrak{B}$, but we do not know $\mathfrak{B}$ itself. Our objective is then to recover a *model* $\mathfrak{B}_S$ *from* S that is an abstraction of $\mathfrak{B}$; i.e. $\mathfrak{B} \subseteq \mathfrak{B}_S$. Clearly, in the recovery process one wants to take into account any structural knowledge of the underlying $\mathfrak{B}$. In the subsequent argument, we focus attention on *time invariant* behaviours $\mathfrak{B}$. [8]

Definition 3.3. Let $\mathfrak{M}_W$ be the set of all time invariant behaviours over W. An experiment $S \subseteq W^*$ is *consistent with time invariance*, if there exists a $\mathfrak{B} \in \mathfrak{M}_W$ such that S is an experiment on $\mathfrak{B}$. Let $\mathfrak{E}_W$ be the set of all experiments over W that are consistent with time invariance. For any $S \in \mathfrak{E}_W$, we say $\mathfrak{B}_S \subseteq W^{\mathbb{N}_0}$ is a *model from* S *under the assumption of time invariance*, if for all $\mathfrak{B} \in \mathfrak{M}_W$:

$$S \text{ is an experiment on } \mathfrak{B} \quad\Rightarrow\quad \mathfrak{B} \subseteq \mathfrak{B}_S\,. \qquad \square$$

Given $S \in \mathfrak{E}_W$, we are most interested in a particularly strong model $\mathfrak{B}_S$, i.e. a small behaviour w.r.t. the partial order "$\subseteq$". We define our candidate as follows:

$$\mathcal{M}^{\downarrow}(S) := \{w \in W^{\mathbb{N}_0} |\; \forall\, k \in \mathbb{N}_0\; \exists\, l \in \mathbb{N}_0 :\; \sigma^k w\big|_{[0,l)} \in S\}.$$

By the below proposition, $\mathcal{M}^{\downarrow}(\,\cdot\,)$ indeed characterises the *strongest model under the assumption of time invariance*:

Proposition 3.4. For any $S \in \mathfrak{E}_W$, the following hold:

(i) $\mathcal{M}^{\downarrow}(S) \in \mathfrak{M}_W$ is a model from S under the assumption of time invariance.

(ii) S is an experiment on $\mathcal{M}^{\downarrow}(S)$.

(iii) $\mathcal{M}^{\downarrow}(S) \subseteq \mathfrak{B}_S$ holds for any model $\mathfrak{B}_S \subseteq W^{\mathbb{N}_0}$ from S the under assumption of time invariance. □

The map $\mathcal{M}^{\downarrow}(\,\cdot\,)$ is interpreted as a parametrisation of the class of behaviours $\mathcal{M}^{\downarrow}(\mathfrak{E}_W) \subseteq \mathfrak{M}_W$, and, in an abstract sense, we say that the behaviour $\mathfrak{B} = \mathcal{M}^{\downarrow}(S)$ is *realised* by the experiment S. Note that, if $|W| \in \mathbb{N}_0$ then the behaviour $\mathfrak{B} = \mathcal{M}^{\downarrow}(S)$ can in fact be realised by a finite automaton. A behaviour $\mathfrak{B}$ is realisable by some experiment $S \in \mathfrak{E}_W$ if and only if there exists an $l \in \mathbb{N}_0$ for which $\mathfrak{B}$ is *l-complete*. [9] Given a behaviour $\mathfrak{B} \in \mathcal{M}^{\downarrow}(\mathfrak{E}_W)$, there exist multiple experiments that realise $\mathfrak{B}$. We ask for a *canonical realisation*, i.e. an experiment $S \in \mathfrak{E}_W$ with $\mathcal{M}^{\downarrow}(S) = \mathfrak{B}$ that is uniquely defined by a distinguishing feature. In the context of hybrid systems, a reasonable objective is to keep the number of strings small and the length of strings short, as this will reduce the computational effort when conducting the experiment. Consistent with the partial order on strings, we define an order relation on experiments [10]

$$\begin{aligned} S_1 \preccurlyeq S_2 \quad :\Leftrightarrow \quad & (\,\forall\, s_2 \in S_2\; \exists\, s_1 \in S_1 :\; s_1 \preccurlyeq s_2\,) \\ & \text{and } (\,\forall\, s_1 \in S_1\; \exists\, s_2 \in S_2 :\; s_1 \preccurlyeq s_2\,)\,. \end{aligned}$$

Indeed, it can be seen that for each $\mathfrak{B} \in \mathcal{M}^{\downarrow}(\mathfrak{E}_W)$ there uniquely exists a *minimal* (w.r.t. "$\preccurlyeq$") experiment $S_{\min} \in \mathfrak{E}_W$ with $\mathfrak{B} = \mathcal{M}^{\downarrow}(S_{\min})$. Moreover, minimal experiments are *prefix-free*. [11]

We now turn to the task of systematic refinement of experiments, which is central to our abstraction based synthesis procedure as developed in Section 2.

[5] W^* denotes the set of finite strings over W; i.e. $W^* := \cup_{l\in\mathbb{N}_0} W^l$.

[6] $|s| \in \mathbb{N}_0$ denotes the length of the finite string $s \in W^*$.

[7] We write $|A| \in \mathbb{N}_0$ to indicate that A is a finite set.

[8] A behaviour $\mathfrak{B}$ is said to be *time invariant* if $\sigma\mathfrak{B} \subseteq \mathfrak{B}$. Here, σ denotes the shift operator; i.e. $\sigma^k : W^{\mathbb{N}_0} \to W^{\mathbb{N}_0}$ is defined by $\sigma^k w(\kappa) := w(\kappa + k)$, $\kappa,\ k \in \mathbb{N}_0$, and $\sigma := \sigma^1$. See [11].

[9] Given $l \in \mathbb{N}_0$, a behaviour $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ is *l-complete* if [$w \in \mathfrak{B} \;\Leftrightarrow\; \forall\, k \in \mathbb{N}_0 :\; \sigma^k w\big|_{[0,l)} \in \mathfrak{B}\big|_{[0,l)}$]; see [11].

[10] The partial order "$\preccurlyeq$" on the set of finite strings W^* is defined by $a \preccurlyeq b$ if and only if there exists $c \in W^*$ such that $b = ac$. We write $a \prec b$ if and only if $a \preccurlyeq b$ and $a \neq b$.

[11] An experiment S is said to be *prefix-free* if ($\forall\, s \in S\; \forall\, \tilde{s} \in W^* :$ if $\tilde{s} \prec s$ then $\tilde{s} \notin S$).

Definition 3.5. An experiment $S_2 \in \mathfrak{E}_W$ is a *refinement* of $S_1 \in \mathfrak{E}_W$ if $\mathcal{M}^{\downarrow}(S_2) \subseteq \mathcal{M}^{\downarrow}(S_1)$. □

For two experiments $S_1, S_2 \in \mathfrak{E}_W$ with $S_1 \preccurlyeq S_2$ it is readily observed that S_2 is a refinement of S_1. However, this condition is not necessary. Given an experiment S_1 on $\mathfrak{B} \in \mathfrak{M}_W$, a refinement S_2 can be generated by replacing strings $s_1 \in S_1$ by longer strings s_2 such that $s_1 \preccurlyeq s_2$. In order to obtain again an experiment on $\mathfrak{B}$, care must be taken to cover all possible future evolution from a string s_1 in the refinement S_2. As a simple example for a sequence of refined experiments, let $S_1 := \mathfrak{B}|_{[0,1)}$ and iteratively define

$$S_{j+1} := \{s \in W^* | \, \exists \tilde{s} \in S_j : \\ \tilde{s} \prec s \text{ and } |s| = |\tilde{s}| + 1 \text{ and } s \in \mathfrak{B}|_{[0,|s|)} \}.$$

Observe that for each j, $S_j \in \mathfrak{E}_W$ is indeed an experiment on $\mathfrak{B}$. Obviously, $S_j \preccurlyeq S_{j+1}$, and hence $\mathfrak{B} \subseteq \mathcal{M}^{\downarrow}(S_{j+1}) \subseteq \mathcal{M}^{\downarrow}(S_j)$ for all $j \in \mathbb{N}_0$. In fact, the generated sequence of models is identical to the so called strongest l-complete approximation; see [7]. One major advantage of the more general framework of experiments is that we still get a refinement if only selected strings are considered, rather than extending every string in an experiment. Consequently, only certain portions of the restricted behaviour $\mathfrak{B}|_{[0,j)}$ need to be computed in each refinement step. The crucial question here is how to determine which strings are worthy of the effort of refinement. Naturally, this depends on the particular problem addressed by the abstraction $\mathcal{M}^{\downarrow}(S_j)$. In what follows, we derive a well motivated notion of strategic refinements for supervisory control.

4. Temporal decomposition of control tasks

We decompose the supervisory control task into two subproblems. A *start up control problem* asks for a controller that drives the plant into a certain mode of operation: closed-loop trajectories are required to initially evolve into a *target* set of finite strings. Once a string in the target is reached, the start up control problem imposes no further performance criteria.

Definition 4.1. Given a plant $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$, a specification $\mathfrak{B}_{\mathrm{spec}} \subseteq W^{\mathbb{N}_0}$, and a *target* $T \subseteq W^*$, let

$$\mathfrak{B}^{\mathrm{stc}}_{\mathrm{spec}} := \{w \in W^{\mathbb{N}_0} | \, \exists l \in \mathbb{N}_0 : \; w|_{[0,l)} \in T \cap \mathfrak{B}_{\mathrm{spec}}|_{[0,l)}\}.$$

A supervisor $\mathfrak{B}^{\mathrm{stc}}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ is said to be a solution of the *start up control problem* $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, T)_{\mathrm{stc}}$ if $\mathfrak{B}^{\mathrm{stc}}_{\mathrm{sup}}$ solves the control problem $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}^{\mathrm{stc}}_{\mathrm{spec}})_{\mathrm{svc}}$. □

The unique existence of a least restrictive solution to the start up control problem follows from Corollary 2.5. If T exhibits a bound on the length of its elements, then $\mathfrak{B}^{\mathrm{stc}}_{\mathrm{spec}}$ is complete. This important case corresponds to targets T that are experiments. Then, the least restrictive solution of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, T)_{\mathrm{stc}}$ is complete.

A *long term control problem* asks for a controller that, once a trajectory evolves into a specified *domain*, restricts all future evolution according to a language inclusion specification. If the closed loop happens *not* evolve into the domain, this *conditional performance criteria* imposes no further restrictions on the closed-loop trajectory.

Definition 4.2. Given a plant $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$, a specification $\mathfrak{B}_{\mathrm{spec}} \subseteq W^{\mathbb{N}_0}$, and a *domain* $D \subseteq W^*$, let

$$\mathfrak{B}^{\mathrm{ltc}}_{\mathrm{spec}} := \{w \in W^{\mathbb{N}_0} | \, \forall \, l \in \mathbb{N}_0 : \\ (w|_{[0,l)} \in D \;\Rightarrow\; w \in \mathfrak{B}_{\mathrm{spec}})\}.$$

A supervisor $\mathfrak{B}^{\mathrm{ltc}}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ is said to be a solution of the *long term control problem* $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, D)_{\mathrm{ltc}}$ if $\mathfrak{B}^{\mathrm{ltc}}_{\mathrm{sup}}$ solves the control problem $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}^{\mathrm{ltc}}_{\mathrm{spec}})_{\mathrm{svc}}$ and if $d \in (\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}^{\mathrm{ltc}}_{\mathrm{sup}})|_{[0,|d|)}$ for all $d \in D$. [12] □

The uniqueness of a least restrictive solution to the long term control problem follows from two observations: (i) the solutions of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}^{\mathrm{ltc}}_{\mathrm{spec}})_{\mathrm{svc}}$ are an upper semi-lattice; and (ii) the extra condition $d \in (\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}^{\mathrm{ltc}}_{\mathrm{sup}})|_{[0,|d|)}$ is retained under unions of long term controllers. The latter *non-triviality condition* rules out the trivial solution $\mathfrak{B}^{\mathrm{ltc}}_{\mathrm{sup}} = \emptyset$ whenever the domain D is nonempty. Hence, for a long term control problem, there is no general guarantee that a solution will exist. This raises the question whether we can at least find a supremal subset $D_{\max} \subseteq D$ on which a solution to $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, D_{\max})_{\mathrm{ltc}}$ does exist. The answer is yes.

Proposition 4.3. Given a complete I/- behaviour $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$, a complete specification $\mathfrak{B}_{\mathrm{spec}} \subseteq W^{\mathbb{N}_0}$, and a prefix-free candidate domain $D \subseteq W^*$ that is an experiment. Then there exists a supremal (w.r.t. "$\subseteq$") subset $D_{\max} \subseteq D$ such that $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, D_{\max})_{\mathrm{ltc}}$ has a solution. We refer to a $D_{\max}$ as the maximal domain of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}}, D)_{\mathrm{ltc}}$. □

We point out an important conceptual feature of our compositional framework. Consider the case where, for the long term control problem, a desired domain can not be achieved. Then we are in the position to weaken our specification to just the maximum domain that can be achieved. This can be interpreted as constructive diagnostic information of what is the best that can be done in the direction of the desired control objective. If this original objective was the ultimate specification in an application context, the diagnosis will not help. However, we continue our framework and decompose the original control problem into start up and long term subproblems. We can then shuffle respon-

[12] Viewed as temporal properties (e.g. [2]), $\mathfrak{B}^{\mathrm{stc}}_{\mathrm{spec}}$ is seen to express an *eventuality* or *guarantee* closed-loop property, while $\mathfrak{B}^{\mathrm{ltc}}_{\mathrm{spec}}$ is the union of the original $\mathfrak{B}_{\mathrm{spec}}$ with a *safety* property (w.r.t. the complement of D).

sibilities between the two controllers and thereby gain valuable hints on *why*, for a particular overall specification, synthesis has failed. This information allows for a much more focused refinement of a plant abstraction than this would be possible from a plain report of failure.

Proposition 4.4. Given a plant $\mathfrak{B}_{\rm p} \subseteq W^{\mathbb{N}_0}$ and a specification $\mathfrak{B}_{\rm spec} \in W^{\mathbb{N}_0}$, let $\mathfrak{B}_{\rm sup}$ denote a nontrivial solution to the control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec})_{\rm svc}$ with closed loop $\mathfrak{B}_{\rm cl} = \mathfrak{B}_{\rm p} \cap \mathfrak{B}_{\rm sup}$. Let $S \subseteq W^*$ be an experiment on $\mathfrak{B}_{\rm cl}$. Then $\mathfrak{B}_{\rm sup}$ is a nontrivial solution to the start up control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec},\ T)_{\rm stc}$ with target $T = S$. Furthermore, $\mathfrak{B}_{\rm sup}$ is a solution to the long term control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec},\ D)_{\rm ltc}$ on the domain $D = S$. □

Proposition 4.4 shows how the overall solution of a control problem can be decomposed into start up and long term components. We now turn to the converse question of how to compose a start up and a long term controller to form an overall solution to a control problem. We define two versions of an operator for the temporal composition of two controllers. In both cases, the switch from one controller to the other is triggered by the trajectory evolving into a certain switching set. For the first operator, the switching condition is tied to the time axis and must be fulfilled from time $k = 0$ onwards.

Definition 4.5. Given two behaviours $\mathfrak{B}^{\rm stc}_{\rm sup}, \mathfrak{B}^{\rm ltc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ and a *switching condition* $C \subseteq W^*$, the combined behaviour $\mathfrak{B}^{\rm stc}_{\rm sup} \wedge^{\circ}_{C} \mathfrak{B}^{\rm ltc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ is defined to be the set of all $w \in W^{\mathbb{N}_0}$ satisfying either (i) or (ii):

(i) there exists an $l \in \mathbb{N}_0$ such that $w|_{[0,l)} \in \mathfrak{B}^{\rm stc}_{\rm sup}|_{[0,l)} \cap C$, and $w \in \mathfrak{B}^{\rm ltc}_{\rm sup}$;

(ii) $w|_{[0,l)} \notin C$ for all $l \in \mathbb{N}_0$, and $w \in \mathfrak{B}^{\rm stc}_{\rm sup}$. □

In contrast, our second operator resets time for the second controller when switching takes place, and here the switching condition does not depend on absolute time.

Definition 4.6. Given two behaviours $\mathfrak{B}^{\rm stc}_{\rm sup}, \mathfrak{B}^{\rm ltc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ and a *switching condition* $C \subseteq W^*$, the combined behaviour $\mathfrak{B}^{\rm stc}_{\rm sup} \wedge_{C} \mathfrak{B}^{\rm ltc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ is defined to be the set of all $w \in W^{\mathbb{N}_0}$ with either (i) or (ii):

(i) there exist $k, l \in \mathbb{N}_0$ such that $w|_{[0,k+l)} \in \mathfrak{B}^{\rm stc}_{\rm sup}|_{[0,k+l)}$, and $\sigma^k w \in \mathfrak{B}^{\rm ltc}_{\rm sup}$, and $\sigma^k w|_{[0,l)} \in C$, and $\sigma^\kappa w|_{[0,\lambda)} \notin C$ for all $\kappa, \lambda \in \mathbb{N}_0$ where $\kappa + \lambda < k + l$;

(ii) $w \in \mathfrak{B}^{\rm stc}_{\rm sup}$ and $\sigma^\kappa w|_{[0,\lambda)} \notin C$ for all $\kappa, \lambda \in \mathbb{N}_0$. □

It can be shown that if both component controllers are generically implementable and/or complete then so are their compositions. Furthermore, the composition operator $\wedge^{\circ}_{C}$ neatly matches our definitions of the start up and long term control problems in that it allows for the composition of an overall solution, provided the target set of the start up controller lies within the domain of the long term controller. This appears an intuitively natural condition for temporal controller composition.

Theorem 4.7. Given a plant $\mathfrak{B}_{\rm p} \subseteq W^{\mathbb{N}_0}$ and a specification $\mathfrak{B}_{\rm spec} \in W^{\mathbb{N}_0}$, let $\mathfrak{B}^{\rm stc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ be a nontrivial solution to the start up control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec},\ T)_{\rm stc}$ for a target $T \subseteq W^*$. Furthermore, let $\mathfrak{B}^{\rm ltc}_{\rm sup} \subseteq W^{\mathbb{N}_0}$ denote a solution to the long term control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec},\ D)_{\rm ltc}$ on a domain $D \subseteq W^*$. If $\mathfrak{B}_{\rm p}$ is a complete I/- behaviour, and if $\mathfrak{B}_{\rm spec}$, $\mathfrak{B}^{\rm stc}_{\rm sup}$ and $\mathfrak{B}^{\rm ltc}_{\rm sup}$ are all complete, and if $T \subseteq D$, then $\mathfrak{B}_{\rm sup} := \mathfrak{B}^{\rm stc}_{\rm sup} \wedge^{\circ}_{T} \mathfrak{B}^{\rm ltc}_{\rm sup}$ is a nontrivial solution to the control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec})_{\rm svc}$. □

5. Strategic experiments

We develop a novel procedure for abstraction based synthesis. In contrast with [7, 8], the main additional feature is that rather than use a global refinement procedure, we focus the refinement according to diagnostic information given when synthesis fails for a particular abstraction. To obtain this diagnostic information, we use the temporal decomposition of the synthesis problem and, in particular, the notion of the maximum domain of the long term control problem. The abstraction procedure is based on experiments and refinement focuses on the complement of the maximum domain of the long term control problem.

Throughout this section, we discuss the supervisory control problem $(\mathfrak{B}_{\rm p},\ \mathfrak{B}_{\rm spec})_{\rm svc}$, subject to the following assumptions, on which we comment below.

(A1) The plant $\mathfrak{B}_{\rm p} \in \mathfrak{M}_W$ is a complete I/- behaviour and the specification $\mathfrak{B}_{\rm spec} \in W^{\mathbb{N}_0}$ is complete.

(A2) There exists an experiment $S \in \mathfrak{E}_W$ on $\mathfrak{B}_{\rm p}$ such that $(\mathcal{M}^{\downarrow}(S),\ \mathfrak{B}_{\rm spec})_{\rm svc}$ exhibits a nontrivial solution.

(A3) We are given a further experiment $\tilde{S} \in \mathfrak{E}_W$ on $\mathfrak{B}_{\rm p}$ such that $\tilde{S} \preccurlyeq S$, and there exist only trivial solutions of the control problem $(\mathcal{M}^{\downarrow}(\tilde{S}),\ \mathfrak{B}_{\rm spec})_{\rm svc}$.

(A4) Let $l_{\rm spec}$ denote the length of a shortest string in $\tilde{S}$; i.e. $l_{\rm spec} := \min\{|\tilde{s}| \mid \tilde{s} \in \tilde{S}\}$. For all $w \in W^*$ such that there exists a $k \in \mathbb{N}_0$ with $w|_{[0,k+l)} \in \mathfrak{B}_{\rm spec}|_{[0,k+l)}$ and $\sigma^k w \in \mathfrak{B}_{\rm spec}$, we require that $w \in \mathfrak{B}_{\rm spec}$.

Ad (A1). If the plant $\mathfrak{B}_{\rm p}$ fails to be complete but is realised by an *I/S/- machine*, we can replace $\mathfrak{B}_{\rm p}$ by its completion [13] and appeal to a variation of Theorem 2.6; see [8, 11]. **Ad (A2) and (A3).** In order to give a constructive discussion of strategic refinement, we require that the control problem can be solved for the abstraction $\mathcal{M}^{\downarrow}(S)$, while we are given another abstraction $\mathcal{M}^{\downarrow}(\tilde{S})$, $\tilde{S} \preccurlyeq S$, on which synthesis fails. In other words: $\mathcal{M}^{\downarrow}(S)$ is, in general, more than sufficiently accurate while $\mathcal{M}^{\downarrow}(\tilde{S})$ is too coarse. Without loss of generality, we may additionally assume that $\tilde{S}$ is prefix-free and

[13] The *completion* of a behaviour $\mathfrak{B}$ is its smallest complete superset, namely $\{w \in W^{\mathbb{N}_0} \mid \forall\, k \in \mathbb{N}_0 :\ w|_{[0,k)} \in \mathfrak{B}|_{[0,k)}\}$.

that $S = \mathfrak{B}_{\mathrm{p}}|_{[0,l)}$ for some $l \in \mathbb{N}_0$. **Ad (A4).** We demand that the strings in $\tilde{S}$ are at least as long as are required for a realisation of $\mathfrak{B}_{\mathrm{spec}}$ by an experiment. Note that this technical requirement is fulfilled for any l_{spec}-complete $\mathfrak{B}_{\mathrm{spec}}$.

Theorem 5.1 gives an experiment $\hat{S}$, $\tilde{S} \preccurlyeq \hat{S} \preccurlyeq S$, that lies between $\tilde{S}$ and S, and that allows for successful supervisory controller synthesis.

Theorem 5.1. Let $\tilde{D}_{\max} \subseteq \tilde{S}$ denote the maximum domain of the long term control problem $(\mathcal{M}^{\downarrow}(\tilde{S}),\ \mathfrak{B}_{\mathrm{spec}},\ \tilde{S})_{\mathrm{ltc}}$. Let

$$\hat{S} := \{s \in S |\ \exists\, \tilde{s} \in \tilde{S} \setminus \tilde{D}_{\max} :\ \tilde{s} \preccurlyeq s\} \cup \tilde{D}_{\max}\,.$$

Under assumptions (A1)–(A4), there exists a nontrivial solution for $(\mathcal{M}^{\downarrow}(\hat{S}),\ \mathfrak{B}_{\mathrm{spec}})_{\mathrm{svc}}$.

Proof (outline). The claim can be established by showing that: (i) $\hat{S}$ is an experiment on $\mathfrak{B}_{\mathrm{p}}$; (ii) $\mathcal{M}^{\downarrow}(S)$, $\mathcal{M}^{\downarrow}(\hat{S})$ and $\mathcal{M}^{\downarrow}(\tilde{S})$ are complete I/- behaviours; (iii) for $\hat{T} := \tilde{D}_{\max} \cup (D \cap \hat{S})$, the start up control problem $(\mathcal{M}^{\downarrow}(\hat{S}),\ \mathfrak{B}_{\mathrm{spec}},\ \hat{T})_{\mathrm{stc}}$ exhibits a nontrivial solution; (iv) for $\hat{D} := \tilde{D}_{\max} \cup (D \cap \hat{S})$, the problem $(\mathcal{M}^{\downarrow}(\hat{S}),\ \mathfrak{B}_{\mathrm{spec}},\ \hat{D})_{\mathrm{ltc}}$ has a nontrivial solution. One then appeals to Theorem 4.7. □

The above theorem suggests the following abstraction based synthesis procedure to solve the supervisory synthesis control problem $(\mathfrak{B}_{\mathrm{p}},\ \mathfrak{B}_{\mathrm{spec}})_{\mathrm{svc}}$, subject to assumptions (A1) and (A2). To keep notation reasonably compact, we assume that $\mathfrak{B}_{\mathrm{spec}}$ is $\hat{l}$-complete for some $\hat{l} \in \mathbb{N}_0$.

(S1) Let $S_0 := \mathfrak{B}_{\mathrm{p}}|_{[0,\hat{l})}$ and $j := 0$

(S2) Compute the least restrictive solution $\mathfrak{B}^{j}_{\mathrm{sup}}$ of $(\mathcal{M}^{\downarrow}(S_j),\ \mathfrak{B}_{\mathrm{spec}})_{\mathrm{svc}}$.

(S3) If $\mathcal{M}^{\downarrow}(S_j) \cap \mathfrak{B}^{j}_{\mathrm{sup}} \neq \emptyset$ terminate this iteration.

(S4) Compute the maximum domain $\tilde{D}^{j}_{\max}$ of $(\mathcal{M}^{\downarrow}(S_j),\ \mathfrak{B}_{\mathrm{spec}},\ S_j)_{\mathrm{ltc}}$ and refine the experiment by

$$S_{j+1} := \{s \in W^* |\ \exists\ \tilde{s} \in S_j \setminus \tilde{D}^{j}_{\max} : \tilde{s} \prec s,\ s \in \mathfrak{B}_{\mathrm{p}}|_{[0,|\tilde{s}|+1)}\} \cup \tilde{D}^{j}_{\max}\,,$$

then proceed with step (S2) for $j := j + 1$.

Note that the synthesis problem in (S2) and the maximum domain in (S4) are not stated for the underlying hybrid plant $\mathfrak{B}_{\mathrm{p}}$, but rather for models from experiments. The latter, in the case of $|W| \in \mathbb{N}$, can be realised by finite automata, and we can perform the computations in (S2) and (S4) by highly efficient algorithms from DES theory. It is readily seen that all S_j are indeed experiments on $\mathfrak{B}_{\mathrm{p}}$. Obviously, S_{j+1} is a refinement of S_j, implementing the strategy to focus on strings that do not lie in the maximal domain.

By (A2), we have assumed that the control problem can be solved via a model based on an experiment. Therefore, we expect that our iteration finds such a suitable model.

Theorem 5.2. Under assumptions (A1) and (A2), the iteration (S1)–(S4) terminates after a finite number of steps. □

6. Conclusions

In many approaches to supervisory controller synthesis for hybrid systems, the major amount of computational effort lies in a reachability analysis of the underlying continuous dynamics. These experiments on the hybrid plant model are used in the construction and in the refinement of a plant abstraction. We decompose our control problem into a start-up part and a long-term part and in doing so we can extract reasons for a failure in the controller synthesis for individual abstractions. From this diagnostic information we strategically *avoid expensive experiments* that are irrelevant to the particular synthesis task at hand. By Theorem 5.2, we demonstrate that there is no loss in our divide-and-conquer strategy: if the synthesis of a nontrivial solution supervisor can be based on some experiment, our method will detect such an experiment. Work in progress includes an efficient implementation of the iteration (S1)–(S4) for a reasonably large class of hybrid systems so that our promising theoretical results can be challenged by application examples.

References

[1] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88, July 2000.

[2] C. Baier and M. Kwiatkowska. On topological hierarchies of temporal properties. *Fundamenta Informaticae*, 41:259–294, 2000.

[3] T. Henzinger. The theory of hybrid automata. In *Proc. of 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 278–292, 1996.

[4] X. Koutsoukos, P. Antsaklis, J. Stiver, and M. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, 88:1026–1049, July 2000.

[5] B. Krogh and A. Chutinan. Hybrid systems: modeling and supervisory control. In P. Frank, editor, *Advances in Control, highlights of ECC'99*, pages 228–246, 1999.

[6] J. Lunze, B. Nixdorf, and H. Richter. Hybrid modelling of continuous-variable systems with application to supervisory control. In *Proc. European Control Conference*, 1997.

[7] T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Systems and Control Letters*, 38:157–166, 1999.

[8] T. Moor, J. Raisch, and S. O'Young. Discrete supervisory control of hybrid systems based on l-complete approximations. *Journal of Discrete Event Dynamic Systems*, 12:83–107, 2002.

[9] P. Philips, M. Weiss, and H. Preisig. Control based on discrete-event models of continuous systems. In *Proc. of the European Control Conference*, 1999.

[10] P. Ramadge and W. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.

[11] J. Willems. Paradigms and puzzles in the theory of dynamic systems. *IEEE Transactions on Automatic Control*, 36:258–294, 1991.