Abstraction-Based Control for Not Necessarily Closed Behaviours

Thomas Moor * **Klaus Schmidt** ** **Thomas Wittmann** *

* *Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg (e-mail: {thomas.moor, thomas.wittmann}@rt.eei.uni-erlangen.de)*
** *Department of Electronic and Communication Engineering, Çankaya University, Ankara (e-mail: schmidt@cankaya.edu.tr)*

Abstract: This paper addresses abstraction-based supervisory control for plant and specification behaviours that are not necessarily ω-closed, i.e. plant behaviours that exhibit eventuality properties and specifications that impose eventuality properties on the closed loop. Technically, the core idea is to combine results from previous work on abstraction-based supervision of input-output behaviours with results on supervisory control of ω-languages. As our main result, we identify a controllability condition for the plant, that ensures a nonblocking closed-loop behaviour with a controller that has been obtained for a plant abstraction.

Keywords: discrete event systems, supervisory control, ω-languages, abstraction-based synthesis

1. INTRODUCTION

The perspective on abstraction-based supervisory control taken in this paper is originally motivated by a class of hybrid control problems, as discussed by, e.g., Cury et al. [1998], Koutsoukos et al. [2000], Moor and Raisch [1999]. Given a plant model with continuous and discrete dynamics, the cited literature proposes methods to obtain a purely discrete plant abstraction with a finite state space. A controller is then designed for the plant abstraction, using variants of supervisory control theory, [Ramadge and Wonham, 1987, 1989]. A crucial question in this approach to hybrid systems is, whether the resulting controller will enforce the desired closed-loop properties not only for the plant abstraction but also for the actual hybrid plant. The affirmative answers given by Cury et al. [1998], Koutsoukos et al. [2000], Moor and Raisch [1999] have in common that the abstraction must account for all possible behaviour of the actual plant. In consequence, *safety properties*, that require the closed loop *not to attain critical configurations*, are retained in an abstraction-based design. In addition, the proposed methods guarantee that the closed loop is *locally* nonblocking, in the sense that at any instance of time controller and plant can agree to execute one more transition. This is a prerequisite, but not a sufficient condition, for a well defined closed-loop behaviour on the infinite time axis. The latter is obtained by additional structural conditions on plant and controller.

In this paper, we extend the results presented by Moor and Raisch [1999] to address general *liveness properties*, i.e., properties that require *desired configurations to be eventually attained*; for a detailed classification of liveness properties in the context of temporal logics, see [Baier and Kwiatkowska, 2000, Manna and Pnueli, 1990]. Technically, liveness properties are related to the notion of ω-closedness: an *ω-language* (language of infinite words) is *ω-closed* if it can be represented as the adherence (infinite extension) of a prefix-closed *$*$-language* (language of finite words). In particular, an ω-closed language can *not* express liveness properties. The structural conditions used by Moor and Raisch [1999] to achieve a nonblocking closed-loop behaviour include an ω-closed plant behaviour and an ω-closed controller. Thus, in the setting of [Moor and Raisch, 1999], neither the plant can exhibit liveness properties, nor can the specification require the controller to impose liveness properties on the closed loop. In this paper we drop the prerequisite of an ω-closed plant and allow for not ω-closed controllers.

As in [Moor and Raisch, 1999], we use a specialized notion of inputs and outputs motivated by behavioural systems theory [Willems, 1991], where the plant must accept any input and the output must not anticipate the input. However, for not necessarily ω-closed behaviours, these conditions turn out too weak in that they fail to ensure a nonblocking closed loop. In this paper, we develop additional requirements that are technically related to the notion of ω-controllability proposed by Thistle and Wonham [1994a]. There, the supervisor must always be in the position to resolve specified closed-loop liveness properties. Our main result is that a particular variation of ω-controllability imposed on the plant itself ensures a nonblocking closed loop, even when the controller has been designed for a plant abstraction. For illustration purposes, we use an academic example with a countable infinite state set. However, our results are immediately applicable to hybrid systems with discrete external behaviour, e.g., in the setting of Moor and Raisch [1999].

This paper is organized as follows. Section 2 summarizes notation and fundamental facts regarding $*$-languages and ω-languages. In Section 3, we recover results from Moor and Raisch [1999] in standard DES notation and identify how it relates to the supervision of ω-languages presented in [Kumar et al., 1992, Ramadge, 1989]. Extensions to not necessarily ω-closed languages are developed in Section 4, on the basis of [Thistle and Wonham, 1994a,b], including our main result on abstraction-based control in the presence of liveness properties.

2. PRELIMINARIES

Let Σ be a *finite alphabet*. The *Kleene-closure* Σ^* is the set of finite strings $s = \sigma_1\sigma_2\cdots\sigma_n$, $n \in \mathbb{N}$, $\sigma_i \in \Sigma$, and the *empty string*

$\epsilon \in \Sigma^*$, $\epsilon \notin \Sigma$. The *length* of a string $s \in \Sigma^*$ is denoted $|s| \in \mathbb{N}_0$, with $|\epsilon| = 0$. A $*$-*language* over Σ is a subset $L \subseteq \Sigma^*$.

If for two strings $s, r \in \Sigma^*$ there exists $t \in \Sigma^*$, $t \neq \epsilon$, such that $s = rt$, we say r is a *strict prefix* of s, and write $r < s$; r is a *prefix* of s if r is a strict prefix of or equal to s and we write $r \leq s$. The prefix of s with length $n \in \mathbb{N}_0$, $n \leq |s|$, is denoted $s^{(n)}$.

The *prefix* of a $*$-language L is defined by $\mathrm{pre}\, L := \{r | \exists\, s \in L : r \leq s\} \subseteq \Sigma^*$. A language L is *prefix-closed* if $L = \mathrm{pre}\, L$. The prefix operator distributes over arbitrary unions of $*$-languages. However, for the intersection of two $*$-languages L and H over Σ, we have $\mathrm{pre}\,(L \cap H) \subseteq (\mathrm{pre}\, L) \cap (\mathrm{pre}\, H)$. If equality holds, L and H are said to be *nonblocking*. This is trivially the case for $H \subseteq L$.

A $*$-language L is *complete* if for all $s \in L$ there exists $\sigma \in \Sigma$ such that $s\sigma \in \mathrm{pre}\, L$; see e.g. [Kumar et al., 1992]. Technically, $L = \emptyset$ is complete. Completeness of a $*$-language must not be confused with *behavioural completeness* as defined by Willems [1991].

Given two $*$-*languages* L and $K \subseteq L$ over Σ and a set of uncontrollable events $\Sigma_u \subseteq \Sigma$, we say K is *controllable* w.r.t. L if $(\mathrm{pre}\, K)\Sigma_u \cap (\mathrm{pre}\, L) \subseteq \mathrm{pre}\, K$. By Ramadge and Wonham [1987, 1989], controllability is retained under arbitrary union: given a family of controllable languages $K_a \subseteq L$, $a \in A$, then the union $K := \cup_{a \in A} K_a$ is controllable, too.

The set of ω-*strings* (countably infinite length strings) over Σ is denoted $\Sigma^\omega := \{w | w = \sigma_1\sigma_2\sigma_3 \cdots$, with $\sigma_i \in \Sigma$ for all $i \in \mathbb{N}\}$. An ω-*language* over Σ is a subset $\mathcal{L} \subseteq \Sigma^\omega$. If for two strings $w \in \Sigma^\omega$, $r \in \Sigma^*$, there exists $v \in \Sigma^\omega$ such that $w = rv$, we say r is a *strict prefix* of w and write $r < w$. The strict prefix of w with length $n \in \mathbb{N}_0$ is denoted $w^{(n)} \in \Sigma^*$. The *prefix* of an ω-language $\mathcal{L} \subseteq \Sigma^\omega$ is defined $\mathrm{pre}\, \mathcal{L} = \{s | \exists\, w \in \mathcal{L} : s < w\} \subseteq \Sigma^*$. Note that, the prefix of any ω-language is complete. The prefix operator distributes over arbitrary unions of ω-languages. However, for the intersection of two ω-languages $\mathcal{L}$ and $\mathcal{H}$ over Σ, we have $\mathrm{pre}\,(\mathcal{L} \cap \mathcal{H}) \subseteq (\mathrm{pre}\, \mathcal{L}) \cap (\mathrm{pre}\, \mathcal{H})$.

Two ω-languages $\mathcal{L}$ and $\mathcal{H}$ over Σ are ω-*nonblocking* if $\mathrm{pre}\,(\mathcal{L} \cap \mathcal{H}) = (\mathrm{pre}\, \mathcal{L}) \cap (\mathrm{pre}\, \mathcal{H})$. Two ω-languages $\mathcal{L}$ and $\mathcal{H}$ over Σ are *locally nonblocking* if $(\mathrm{pre}\, \mathcal{L}) \cap (\mathrm{pre}\, \mathcal{H})$ is complete. Any two languages that are ω-nonblocking are also locally nonblocking. Note that for $\mathcal{H} \subseteq \mathcal{L}$ both nonblocking conditions are trivially satisfied.

For a language $L \subseteq \Sigma^*$, the *adherence* is defined $\mathrm{adh}\, L = \{w \in \Sigma^\omega | \forall\, n \in \mathbb{N}_0\ \exists r \in \Sigma^* : w^{(n)} r \in L\}$; see e.g. [Boasson and Nivat, 1980]. If and only if a $*$-language $L \subseteq \Sigma^*$ is complete and prefix-closed, we have $\mathrm{pre}\, \mathrm{adh}\, L = L$; see [Kumar et al., 1992].

The ω-*closure* of an ω-language $\mathcal{L}$ is defined by $\mathrm{clo}\, \mathcal{L} := \mathrm{adh}\, \mathrm{pre}\, \mathcal{L}$. This definition is equivalent to the *topological closure* w.r.t. the topology induced by the metric $d\colon \Sigma^\omega \times \Sigma^\omega \to \mathbb{R}_0^+$, with $d(w, v) := (\frac{1}{2})\hat{}\min\{n \in \mathbb{N} | w^{(n)} \neq v^{(n)}\}$ for $w \neq v$ and $d(w, w) = 0$, respectively; see [Boasson and Nivat, 1980]. An ω-language $\mathcal{L}$ is said to be ω-*closed* if $\mathrm{clo}\, \mathcal{L} = \mathcal{L}$, i.e., if $\mathrm{adh}\, \mathrm{pre}\, \mathcal{L} = \mathcal{L}$. In the context of behavioural systems theory, ω-closedness is referred to as *behavioural completeness*; see, e.g., [Willems, 1991], Definition II.4. The adherence of a prefix-closed $*$-language is ω-closed. Given two ω-languages $\mathcal{L}$ and $\mathcal{H}$, we say $\mathcal{L}$ is *relatively* ω-*closed w.r.t.* $\mathcal{H}$ if $\mathcal{L} = (\mathrm{clo}\, \mathcal{L}) \cap \mathcal{H}$. The closure operator distributes over finite unions of ω-languages. However, for an arbitrary family of ω-languages $\mathcal{L}_a$, $a \in A$, we have $\cup_{a \in A} \mathrm{clo}\, \mathcal{L}_a \subseteq \mathrm{clo}\, (\cup_{a \in A} \mathcal{L}_a)$.

Provided that two ω-languages $\mathcal{L}$ and $\mathcal{H}$ are locally nonblocking, $\mathrm{clo}\,(\mathcal{L} \cap \mathcal{H}) = (\mathrm{clo}\, \mathcal{L}) \cap (\mathrm{clo}\, \mathcal{H})$ is equivalent to the two languages to be ω-nonblocking. In particular, for ω-closed languages, the local nonblocking property is equivalent to the ω-nonblocking property.

3. ABSTRACTION-BASED CONTROLLER SYNTHESIS

Given a plant and a plant abstraction, we ask for conditions, under which controller synthesis can be carried out based on the abstraction while the resulting controller is guaranteed to enforce desired closed-loop properties for the actual plant. In our discussion, we model the plant, its abstraction and the controller by ω-languages and we require the closed loop to satisfy a language inclusion specification. In this section, we only address the local behaviour, which is characterized by the prefixes of the respective languages, or, equivalently, by their ω-closure. As a structural liveness property, we require the closed loop to be locally nonblocking.

In order to facilitate the comparison of solutions to different control problems, we refer to the following formal definition:

Definition 1. A *control problem* is a tuple $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, where Σ is the overall alphabet, $\Sigma_u \subseteq \Sigma$ is the set of *uncontrollable events*, $\mathcal{L} \subseteq \Sigma^\omega$ is the *plant behaviour* and $\mathcal{E} \subseteq \Sigma^\omega$ is the *specification*. A *controller* $\mathcal{H} \subseteq \Sigma^\omega$ is *locally admissible* if

(i) $(\mathrm{pre}\, \mathcal{L}) \cap (\mathrm{pre}\, \mathcal{H})$ is controllable w.r.t. $\mathrm{pre}\, \mathcal{L}$; and
(ii) $\mathcal{L}$ and $\mathcal{H}$ are locally nonblocking.

A controller $\mathcal{H} \subseteq \Sigma^\omega$ is a *local solution* if it is locally admissible and if it *locally enforces the specification*:

(iii) $(\mathrm{pre}\, \mathcal{L}) \cap (\mathrm{pre}\, \mathcal{H}) \subseteq \mathrm{pre}\, \mathcal{E}$. □

Our notion of a control problem and its solutions refers to the plant and controller dynamics and, thereby, implicitly imposes conditions on the closed-loop behaviour. This contrasts the common approach taken in supervisory control theory, where the controller is modelled by a supervisor map to implement a causal feedback. However, at this stage, the difference is purely cosmetic and we can recover the existence of a least restrictive solution in analogy to the established theory.

Proposition 2. Given a control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, the set of all local solutions is non-empty and forms a complete upper semi-lattice w.r.t. set-inclusion. □

According to the above proposition, a control problem exhibits a uniquely defined supremal solution, in the following denoted $\mathcal{H}^\uparrow$. Since the language inclusion specification in Definition 1, condition (iii), refers to the prefix only, and, thus, does not impose a liveness property on the closed loop, $\mathcal{H}^\uparrow$ turns out ω-closed.

Proposition 3. If $\mathcal{H}$ is a local solution to the control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, then so is $\mathrm{clo}\, \mathcal{H}$. In particular, the supremal local solution $\mathcal{H}^\uparrow$ is ω-closed. □

As a consequence, for an ω-closed plant $\mathcal{L}$, the closed loop under minimal restrictive control $\mathcal{H}^\uparrow$ is ω-nonblocking. In this case, the intersection $\mathcal{K}^\uparrow := \mathcal{L} \cap \mathcal{H}^\uparrow$ is an adequate model of the infinite time closed-loop behaviour. The following proposition generalizes this observation. It relates our notion of a solution to a control problem to the infinite-time controllability condition proposed by Ramadge [1989].

Proposition 4. Let $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$ be a control problem where the specification $\mathcal{E}$ is relatively closed w.r.t. $\mathcal{L}$. If an ω-closed local

solution $\mathcal{H}$ and the plant $\mathcal{L}$ are ω-nonblocking, then the closed-loop $\mathcal{K} := \mathcal{L} \cap \mathcal{H}$ possesses the following properties:

(i) $\mathrm{pre}\,\mathcal{K}$ is controllable w.r.t. $\mathrm{pre}\,\mathcal{L}$;
(ii) $\mathcal{K}$ is relatively ω-closed w.r.t. $\mathcal{L}$; and
(iii) $\mathcal{K} \subseteq \mathcal{E}$.

Vice versa, for any ω-language $\mathcal{K}$, that satisfies conditions (i)–(iii), $\mathcal{H} = \mathrm{clo}\,\mathcal{K}$ is a local solution. □

In particular, if the plant $\mathcal{L}$ and the specification $\mathcal{E}$ are ω-closed, the prerequisites of the above proposition can always be satisfied by using the minimal restrictive solution $\mathcal{H}^{\uparrow}$. Referring to [Ramadge, 1989], Proposition 3.1, properties (i) and (ii) then guarantee the existence of a supervisor map $f\colon \Sigma^* \to \Gamma$, $\Gamma := \{\gamma \subseteq \Sigma | \Sigma_u \subseteq \gamma\}$ to implement a causal feedback that enforces the closed-loop behaviour $\mathcal{K}^{\uparrow} := \mathcal{L} \cap \mathcal{H}^{\uparrow}$; see Figure 1. For the case that the plant and the specification are realized by Büchi automata, Kumar et al. [1992] propose algorithms for the computation of a transition system that implements $\mathcal{K}^{\uparrow}$; for a software implementation see e.g. libFAUDES [2006–2011].

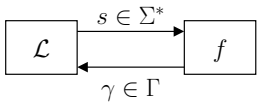

Fig. 1. Closed loop with supervisor map f

We now address the situation where the controller design is based on a plant abstraction $\mathcal{L}'$ that accounts for all possible behaviour of the actual plant $\mathcal{L}$. In our discussion, we express the latter requirement by the inclusion $\mathcal{L} \subseteq \mathcal{L}'$. The inclusion is motivated by hybrid systems, e.g., in the setting of [Cury et al., 1998] or [Moor and Raisch, 1999].

It is easily verified that any solution $\mathcal{H}$ to the control problem $(\Sigma, \Sigma_u, \mathcal{L}', \mathcal{E})$ when used as a controller for the actual plant $\mathcal{L}$, also satisfies conditions (i) and (iii) of Definition 1:

$$\begin{aligned} & ((\mathrm{pre}\,\mathcal{L}) \cap (\mathrm{pre}\,\mathcal{H}))\Sigma_u \cap (\mathrm{pre}\,\mathcal{L}) \\ \subseteq\ & ((\mathrm{pre}\,\mathcal{L}') \cap (\mathrm{pre}\,\mathcal{H}))\Sigma_u \cap (\mathrm{pre}\,\mathcal{L}') \cap (\mathrm{pre}\,\mathcal{L}) \\ \subseteq\ & (\mathrm{pre}\,\mathcal{L}') \cap (\mathrm{pre}\,\mathcal{H}) \cap (\mathrm{pre}\,\mathcal{L}) \\ =\ & (\mathrm{pre}\,\mathcal{L}) \cap (\mathrm{pre}\,\mathcal{H}), \end{aligned}$$

and

$$(\mathrm{pre}\,\mathcal{L}) \cap (\mathrm{pre}\,\mathcal{H}) \ \subseteq\ (\mathrm{pre}\,\mathcal{L}') \cap (\mathrm{pre}\,\mathcal{H}) \ \subseteq\ \mathrm{pre}\,\mathcal{E}.$$

However, the nonblocking property of Definition 1, (ii), in general will not be satisfied. In [Moor and Raisch, 1999], a variation of input-output behaviours [Willems, 1991] is used to ensure a structural liveness property. There, input and output symbols alternate, the plant accepts any input signal and the controller accepts any output signal. This notion of inputs and outputs is motivated by sampled-data continuous systems and, in contrast to Figure 1, leads to the closed-loop configuration illustrated by Figure 2: rather than to apply a control pattern that enables a set of events, the controller here applies a particular input symbol in order to receive the next output symbol.

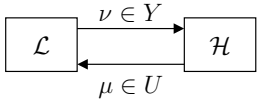

Fig. 2. Input/output system interconnection

The following definition restates two conditions from Moor and Raisch [1999] that, when imposed on plant and controller, justify the above interpretation of a closed-loop configuration.

Definition 5. Let $\mathcal{L}$ be an ω-language over Σ with input symbols $U \subseteq \Sigma$. Then

(a) $\mathcal{L}$ has a *locally free input* U, if
$(\forall\ s \in \Sigma^*,\ \mu \in U,\ \mu' \in U)[\, s\mu \in \mathrm{pre}\,\mathcal{L} \ \Rightarrow\ s\mu' \in \mathrm{pre}\,\mathcal{L}\,]$;
(b) $\mathcal{L}$ has a *relatively locally free input* U *w.r.t.* $\mathcal{K} \subseteq \Sigma^{\omega}$, if
$(\forall\ s \in \Sigma^*,\ \mu \in U)[\, s\mu \in \mathrm{pre}\,\mathcal{K},\ s \in \mathrm{pre}\,\mathcal{L} \ \Rightarrow\ s\mu \in \mathrm{pre}\,\mathcal{L}\,]$. □

A system with behaviour $\mathcal{L}$ that satisfies condition (a) will at any instance of time either accept any input symbol or no input symbol at all. In the situation of alternating input and output symbols $\mathcal{L} \subseteq (UY)^{\omega}$, $\Sigma = U \dot\cup Y$, and if $\mathcal{L}$ is ω-closed and non-empty, condition (a) is equivalent to a *free input* and a *non-anticipating output*; see [Willems, 1991], Definition VIII.3, conditions (1) and (2). We will impose condition (a) on the plant behaviour. On the other hand, we require the controller $\mathcal{H} \subseteq (UY)^{\omega}$ to exhibit a relatively locally free input Y w.r.t. the plant $\mathcal{L}$. In this setting, (b) amounts to a controllability condition where Y is regarded the set of uncontrollable events. With both conditions in place, we obtain a closed-loop behaviour with alternating input and output symbols that are locally accepted by the plant and the controller, respectively. As intended, the closed loop turns out locally nonblocking.

Proposition 6. Let $\Sigma = U \dot\cup Y$ be an alphabet partitioned in input symbols and output symbols U and Y, respectively, and consider a plant $\mathcal{L} \subseteq (UY)^{\omega}$ and a controller $\mathcal{H} \subseteq (UY)^{\omega}$. If $\mathcal{L}$ has a locally free input U and if $\mathcal{H}$ has a relatively locally free input Y w.r.t. $\mathcal{L}$, then $\mathcal{L}$ and $\mathcal{H}$ are locally nonblocking. □

Remark 7. The prerequisite $\mathcal{L}, \mathcal{H} \subseteq (UY)^{\omega}$ ensures not only that input symbols and output symbols alternate, but also that the sequence starts with an input symbol. This choice was made deliberately: a corresponding proposition holds when all sequences start with an output symbol, i.e. $\mathcal{L}, \mathcal{H} \subseteq (YU)^{\omega}$. □

The following theorem summarizes the results for abstraction-based controller synthesis obtained in this section.

Theorem 8. Let $\Sigma = U \dot\cup Y$ be an alphabet partitioned in input symbols and output symbols U and Y, respectively, and consider a plant $\mathcal{L} \subseteq (UY)^{\omega}$ with locally free input U and a plant abstraction $\mathcal{L}' \subseteq (UY)^{\omega}$, $\mathcal{L} \subseteq \mathcal{L}'$. Then any local solution $\mathcal{H}$ of the control problem $(\Sigma, Y, \mathcal{L}', \mathcal{E})$ is also a local solution of $(\Sigma, Y, \mathcal{L}, \mathcal{E})$. □

If in addition to the hypothesis of Theorem 8 the plant and the specification are ω-closed, the supremal local solution $\mathcal{H}^{\uparrow}$ to $(\Sigma, Y, \mathcal{L}', \mathcal{E})$ and $\mathcal{L}$ are ω-nonblocking. In this case, the infinite time behaviour of the closed loop is modelled by $\mathcal{K}^{\uparrow} := \mathcal{L} \cap \mathcal{H}^{\uparrow}$ and satisfies the specification w.r.t. infinite time, i.e. $\mathcal{K}^{\uparrow} \subseteq \mathcal{E}$.

We illustrate our results by means of an academic example. Consider a vehicle on a path that can be controlled by input events `e` for *"move to the east"* and `w` for *"move to the west"*, respectively. A measurement facility distinguishes a home position and issues output events `H` for *"home"*, `E` for *"east of home"* or `W` for *"west of home"*. Thus, the relevant alphabets are given by $U = \{\mathtt{e}, \mathtt{w}\}$, $Y = \{\mathtt{H}, \mathtt{E}, \mathtt{W}\}$, $\Sigma = U \dot\cup Y$. The dynamics of the vehicle are modelled by the transition system given in Figure 3, where we use an infinite state space to represent an unbounded path. For now, we restrict the initial state to the home position `H0`. The plant behaviour $\mathcal{L}$ is the set of infinite words generated by the transition system starting from `H0`. In the absence of further acceptance conditions, the unique initial state and the determinism of the transition system imply that $\mathcal{L}$ is ω-closed. By inspection, U is verified to be a locally free input.

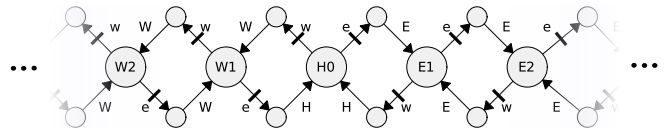

Fig. 3. Plant model $\mathcal{L}$

Since state space enumeration is not feasible for an infinite state set, we can not directly apply computational methods from supervisory control theory. Thus, we resort to a plant abstraction with finite state set. The abstraction $\mathcal{L}'$, realized by the transition system in Figure 4, has been obtained by aggregating all states east of position `E1` and all states west of position `W1`. It corresponds to the so called l-complete abstraction with $l = 1$, as proposed by Moor and Raisch [1999], and, consequently, we have indeed $\mathcal{L} \subseteq \mathcal{L}'$.

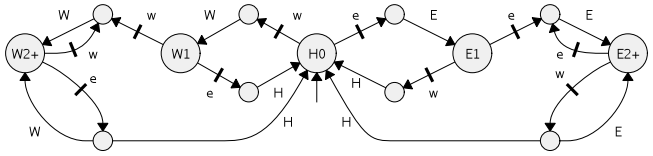

Fig. 4. Plant abstraction $\mathcal{L}'$

We want to design a controller that enforces the vehicle to regularly visit east and west positions. At this stage, we circumvent eventuality and use the more restrictive formal specification Figure 5 and thereby require the vehicle to alternate between the positions `E1` and `W1`. A controller has been obtained by applying the algorithms presented by Kumar et al. [1992] to the abstraction $\mathcal{L}'$; see Figure 6. According to Theorem 8, the controller is also a local solution for the actual plant. By ω-closedness of plant and specification, the closed loop is ω-nonblocking and satisfies the specification w.r.t. infinite time.

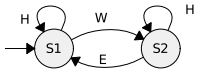

Fig. 5. Specification $\mathcal{E}$, projected onto $\{\mathtt{H}, \mathtt{E}, \mathtt{W}\}^\omega$

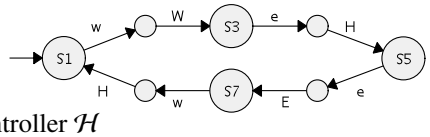

Fig. 6. Controller $\mathcal{H}$

The example was intentionally chosen to demonstrate the limitations that arise from restricting the discussion to ω-closed behaviours. In particular, the formal specification was overly restrictive and we required the plant initial state to be known. We will continue the example at the end of the following section to illustrate our results on abstraction-based controller synthesis for not necessarily ω-closed behaviours.

4. NOT NECESSARILY CLOSED BEHAVIOURS

For ω-closed behaviours, we used a variation of input-output behaviours to establish nonblocking abstraction-based control. In order to address situations where the plant and/or the specification fail to be ω-closed, stronger conditions are required that account for the liveness properties modelled by the plant or enforced by the controller. Such conditions have been proposed by Wittmann [2010], where the technical dicussion is based on a particular union representation of not ω-closed languages. In this section, we follow a different approach in that we define the notion of ω-admissibility as a variation of ω-controllability [1] and thereby relate our results to the work of Thistle and Wonham [1994a,b]. As it turns out, the union representation from Wittmann [2010] is recovered by Proposition 13 below.

Definition 9. Given a control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, a controller $\mathcal{H} \subseteq \Sigma^\omega$ is *ω-admissible* if for all $s \in (\mathrm{pre}\,\mathcal{L}) \cap (\mathrm{pre}\,\mathcal{H})$ there exists $\mathcal{V}_s \subseteq \mathcal{L} \cap \mathcal{H}$ with $s \in \mathrm{pre}\,\mathcal{V}_s$ and

(i) $\mathrm{pre}\,\mathcal{V}_s$ is controllable w.r.t. $\mathrm{pre}\,\mathcal{L}$; and
(ii) $\mathcal{V}_s$ is relatively ω-closed w.r.t. $\mathcal{L}$.

A controller $\mathcal{H} \subseteq \Sigma^\omega$ is an *ω-solution* if it is ω-admissible and if it *enforces the specification for infinite time*:

(iii) $\mathcal{L} \cap \mathcal{H} \subseteq \mathcal{E}$. □

Conditions (i) and (ii) imposed on $\mathcal{V}_s$ conform with the notion of infinite-time controllability proposed by Ramadge [1989]. In particular, for each $\mathcal{V}_s$, there exists a supervisor map $f : \Sigma^* \to \Gamma$, to implement a causal feedback that enforces the closed-loop behaviour $\mathcal{V}_s$. By quantification over all prefixes s from the local closed loop $(\mathrm{pre}\,\mathcal{L}) \cap (\mathrm{pre}\,\mathcal{H})$, the above definition requires the persistent existence of a supervisor map that can take over to run the plant for infinite time within $\mathcal{L} \cap \mathcal{H}$. This is essentially the same requirement as imposed by ω-controllability, according to the definition of Thistle and Wonham [1994a]. Indeed, for the case of $\mathcal{H} \subseteq \mathcal{L}$, ω-admissibility can be verified to be equivalent to ω-controllability. However, in the context of abstraction-based control, we also need to address the situation where $\mathcal{H} \not\subseteq \mathcal{L}$.

By the following proposition, ω-admissibility implies local admissibility and an ω-nonblocking closed loop. In particular, for an ω-solution $\mathcal{H}$ to a control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$ the intersection $\mathcal{L} \cap \mathcal{H}$ indeed is an adequate model of the closed-loop configuration and this justifies the terminology of Definition 9, part (iii).

Proposition 10. Given a control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, let $\mathcal{H}$ denote an ω-admissible controller. Then $\mathcal{H}$ is locally admissible, $\mathcal{L}$ and $\mathcal{H}$ are ω-nonblocking and the closed loop $\mathcal{K} = \mathcal{L} \cap \mathcal{H}$ is ω-admissible. □

Recall that, for ω-closed languages, local admissibility implies ω-nonblockingness. Under the same hypothesis and refering to Definition 9, we can choose $\mathcal{V}_s = \mathcal{L} \cap \mathcal{H}$ to observe ω-admissibility. Together with the above proposition we conclude that ω-admissibility and local admissibility are equivalent for ω-closed languages.

As with local admissibility, ω-admissibility is retained under arbitrary union. Thus, any control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$ exhibits a supremal ω-solution $\mathcal{H}^\Uparrow$ and, in turn, a supremal closed-loop behaviour $\mathcal{K}^\Uparrow = \mathcal{L} \cap \mathcal{H}^\Uparrow$.

Proposition 11. Given a control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, the set of all ω-solutions is non-empty and forms a complete upper semi-lattice w.r.t. set-inclusion. □

The following proposition characterises the closed-loop behaviour under minimal restrictive control $\mathcal{H}^\Uparrow$ as the union of all closed-loop behaviours of local solutions that are ω-nonblocking and enforce the specification on infinite time.

[1] The terminology *ω-controllability*, as defined by Thistle and Wonham [1994a], must not be confused with the requirements for infinite-time controllability proposed in Ramadge [1989]. In the context of this paper, *ω-controllability* is always understood in the sense of Thistle and Wonham [1994a].

Proposition 12. Given the control problem $(\Sigma, \Sigma_u, \mathcal{L}, \mathcal{E})$, denote the supremal ω-solution $\mathcal{H}^\Uparrow$ with supremal closed-loop behaviour $\mathcal{K}^\Uparrow = \mathcal{L} \cap \mathcal{H}^\Uparrow$. Then

$$\mathcal{K}^\Uparrow = \cup \{\mathcal{K} \subseteq \mathcal{E} | \quad \text{pre}\,\mathcal{K} \text{ is controllable w.r.t. pre}\,\mathcal{L}\,, \text{ and } \mathcal{K} \text{ is relatively closed w.r.t. } \mathcal{L}\}\,. \qquad \square$$

By the above union representation, $\mathcal{K}^\Uparrow$ is identical to the supremal ω-controllable sublanguage as characterized in [Thistle and Wonham, 1994a], Corollary 5.4. In particular, one may use the algorithms presented in [Thistle and Wonham, 1994a,b] to compute a realisation of $\mathcal{K}^\Uparrow$ based on suitable realisations of $\mathcal{L}$ and $\mathcal{E}$. An experimental software implementation for Büchi automata is available within libFAUDES [2006–2011].

For the purpose of abstraction-based control, consider an ω-solution $\mathcal{H}$ to the control problem $(\Sigma, \Sigma_u, \mathcal{L}', \mathcal{E})$ where $\mathcal{L}'$ is an abstraction of the actual plant $\mathcal{L}$, i.e. $\mathcal{L} \subseteq \mathcal{L}'$. By $\mathcal{L} \cap \mathcal{H} \subseteq \mathcal{L}' \cap \mathcal{H} \subseteq \mathcal{E}$, the controller $\mathcal{H}$ enforces the specification for infinite time when applied to the actual plant $\mathcal{L}$, and we are left to establish conditions for ω-admissibility. Motivated by Theorem 8 for ω-closed behaviours, we partition the overall alphabet in input symbols and output symbols and require them to alternate, i.e. $\Sigma = U \dot\cup Y$, $\mathcal{L} \subseteq (UY)^\omega$. To observe that one can *not* expect a locally free input of the plant to be a sufficient condition for an ω-nonblocking closed-loop configuration, consider the transition system given in Figure 7. Here, we interpret the marked states according to Büchi's acceptance condition: an infinite execution sequence is accepted if it passes infinitely often some marked state. Clearly, the corresponding ω-language $\mathcal{L}$ has a locally free input U. Furthermore, $\mathcal{L}$ can be verified to satisfy the behavioural conditions for a *free input* and a *non-anticipating output*; see [Willems, 1991], Definition VIII.3, conditions (1) and (2). However, the system must eventually exit the states `A` or `B` by the input symbol `b` or `a`, respectively, to reach a marked state. Thus, $\mathcal{L}$ exhibits a liveness property that restricts the input on the infinite time axis in relation to the output. If a controller design is based on an abstraction, e.g. $\mathcal{L}' := \text{clo}\,\mathcal{L}$, the synthesis procedure may not respect the liveness property of the actual plant. For example, the controller $\mathcal{H} = \mathtt{a}\{\mathtt{Aa}, \mathtt{Bb}\}^\omega$ is ω-admissible to the abstraction $\mathcal{L}'$, but conflicts with the acceptance condition of the actual plant $\mathcal{L}$.

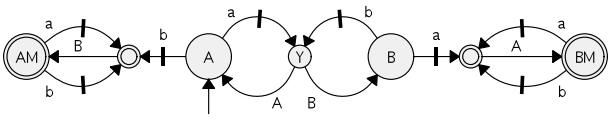

Fig. 7. Eventuality property imposed on the input

To prevent this conflict situation, we require the plant to be always in the position to choose its outputs such that it satisfies its own liveness properties and to do so independently of the future inputs. This can be formally expressed as a controllability condition where the input symbols are regarded the uncontrollable events. In summary, we impose the following conditions on the actual plant $\mathcal{L}$:

A1 $\mathcal{L}$ has a locally free input U ; and
A2 $\mathcal{L}$ is an ω-solution to the control problem $(\Sigma, U, \text{clo}\,\mathcal{L}, \mathcal{L})$.

Note that, for ω-closed plant behaviours $\mathcal{L}$, A2 is trivially fulfilled, and in this sense the structural conditions A1 and A2 are a generalisation of the approach taken in the previous section for ω-closed plants.

Furthermore, if a plant $\mathcal{L}$ satisfies A2, the corresponding formal closed-loop behaviour amounts to $(\text{clo}\,\mathcal{L}) \cap \mathcal{L} = \mathcal{L}$. Since, in A2, $\mathcal{L}$ also plays the role of the specification, we must have $\mathcal{L} = (\text{clo}\,\mathcal{L}) \cap \mathcal{H}^\Uparrow = \mathcal{K}^\Uparrow$, where $\mathcal{H}^\Uparrow$ denotes the supremal ω-solution of $(\Sigma, U, \text{clo}\,\mathcal{L}, \mathcal{L})$. When $\mathcal{L}$ is realized by a finite transition system, A2 can be verified by essentially the same procedures as those used for the computation of the supremal ω-solution.

The following proposition states an alternative characterization of A1 and A2. It turns out useful for the verification of A1 and A2 for infinite-state systems when eventuality properties are encoded by an unknown initial state. This is the case for the hybrid systems discussed in [Moor et al., 2002] as well as for the example presented at the end of this section.

Proposition 13. Let $\mathcal{L}$ be an ω-language over Σ with input symbols $U \subseteq \Sigma$. Then $\mathcal{L}$ satisfies A1 and A2 if and only if there exists a family $\mathcal{K}_a$, $a \in A$, such that $\mathcal{L} = \cup_{a \in A} \mathcal{K}_a$, where, for all $a \in A$

(a) $\mathcal{K}_a$ has a locally free input U, and
(b) $\mathcal{K}_a$ is ω-closed. $\square$

Our main result is the following generalisation of Theorem 8 for not necessarily ω-closed behaviours.

Theorem 14. Let $\Sigma = U \dot\cup Y$ be an alphabet partitioned in input symbols and output symbols U and Y, respectively. Let the plant $\mathcal{L} \subseteq (UY)^\omega$ satisfy the requirements A1 and A2, and consider a plant abstraction $\mathcal{L}' \subseteq (UY)^\omega$, $\mathcal{L} \subseteq \mathcal{L}'$. Then any ω-solution $\mathcal{H}$ to the control problem $(\Sigma, Y, \mathcal{L}', \mathcal{E})$ is also an ω-solution to $(\Sigma, Y, \mathcal{L}, \mathcal{E})$. $\square$

We continue the example from Section 4, still referring to the transition system in Figure 3 as our plant model. However, we do not restrict the initial state to any particular value except that an input event must be accepted. To observe that the resulting plant behaviour $\mathcal{L}$ is not ω-closed, consider an infinite number of `w` input symbols events to be applied and assume that the first output symbol is `E`. Thus, the vehicle starts at some east position and will *eventually* reach the home position. However, the number of transitions required is unbounded and we have $(\mathtt{wE})^* \subseteq \text{pre}\,\mathcal{L}$ while $\text{adh}\,(\mathtt{wE})^* \not\subseteq \mathcal{L}$, and, hence, $(\mathtt{wE})^\omega \in \text{clo}\,\mathcal{L}$ but $(\mathtt{wE})^\omega \notin \mathcal{L}$. This example illustrates that, even in the absence of any particular acceptance condition, an unknown initial state within an infinite range can lead to a not ω-closed behaviour. This may be regarded as an interpretation of how eventuality properties are implemented by a physical plant.

Regarding conditions A1 and A2, we represent $\mathcal{L}$ as the union over all languages $\mathcal{L}_q$, $q \in Q$, generated by the transition system Figure 3, where the initial state is parametrized by q and Q denotes the overall state set. Determinism of the transition system and the unique initial state imply that each $\mathcal{L}_q$ is ω-closed. The free input U is verified by inspection, and, hence, conditions A1 and A2 are implied by Proposition 13.

To express liveness properties in the plant abstraction, we use marked states and interpret the transition system w.r.t. Büchi's acceptance condition. The particular abstraction $\mathcal{L}'$, Figure 8, has been obtained by merging all east and west positions and by choosing a marking that expresses the plant property that when starting in a west position and driving continuously east, the vehicle will eventually pass the home position. The abstraction $\mathcal{L}'$ does, however, not possess the original plant property that driving two steps to the east and one to the west will also eventually lead to the home position; i.e., we have $\mathcal{L}' \supsetneq \mathcal{L}$.

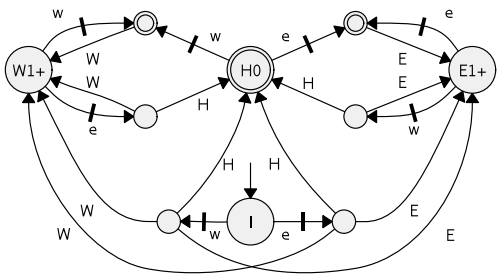

Fig. 8. Plant abstraction $\mathcal{L}'$ with liveness properties

In our specification, Figure 9, we require that an east position is eventually followed by a west position and vice versa. The below controller has been obtained based on the plant abstraction $\mathcal{L}'$ using the software implementation within libFAUDES [2006–2011]. By Theorem 14, it also is an ω-solution to the control problem for the actual plant. In particular, $\mathcal{L}$ and $\mathcal{H}$ are ω-nonblocking and $\mathcal{K} = \mathcal{L} \cap \mathcal{H}$ represents the closed-loop behaviour. Regarding the respective transition systems, $\mathcal{K}$ consists of all infinite strings that pass marked states of both, plant and controller, infinitely often. Note that this contrasts the common interpretation for $*$-languages, realized by finite state automata, where marked states must be attained simultaneously.

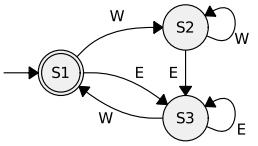

Fig. 9. Specification $\mathcal{E}$, projected onto $\{\mathtt{E}, \mathtt{W}\}^\omega$

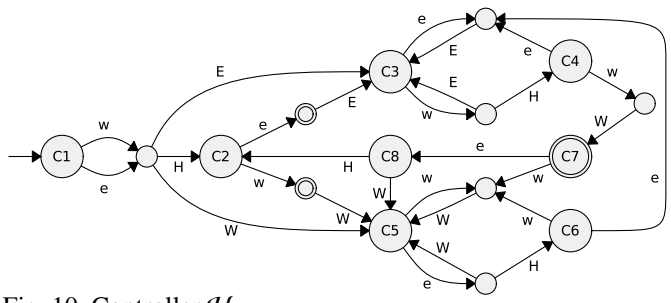

Fig. 10. Controller $\mathcal{H}$

In contrast to Section 3, the controller here allows the vehicle to visit any position. It can, for example, move arbitrarily far to the east by executing any finite sequence of `eE` events in state `C3`. Since the respective states are not marked, it can, however, not execute `eE` forever. Thus, when in state C3 the controller must eventually move to the west. If it does so sufficiently long, the according `wE` sequence will be eventually followed by `wH` to attain state `C4`. This liveness property is expressed by the plant abstraction and it is present in the actual plant. From `C4`, the controller can reach a marked state `C7` by a subsequent `wW`, finally visiting a west position in compliance with the specification. Regarding a physical implementation of the controller $\mathcal{H}$, we may refer to the union representation of Proposition 12 and resolve eventuality properties of $\mathcal{H}$ by choosing one particular component as basis of an implementation.

5. CONCLUSION

We identified conditions for abstraction-based controller synthesis in the presence of liveness properties that guarantee a well defined closed-loop behaviour on the infinite time axis. Technically, the proposed conditions are built on the notion of ω-controllability, as introduced in [Thistle and Wonham, 1994a]. In order to guarantee an ω-nonblocking closed loop, the conditions are slightly stronger than free inputs and non-anticipating outputs used in [Moor and Raisch, 1999]. The perspective we take is motivated by the supervision of hybrid systems and our results are directly applicable in the context of e.g. [Moor and Raisch, 1999, Cury et al., 1998], provided that a discrete event abstraction can be obtained. In ongoing work, we address the computation of finite state abstractions that maintain relevant liveness properties. The focus there is on linear hybrid automata, where specific liveness properties can be characterized by sets of continuous states that are invariant under affine transformations obtained from reachability analysis.

REFERENCES

C. Baier and M.Z. Kwiatkowska. On topological hierarchies of temporal properties. *Fund. Inform.*, 41:259–294, 2000.

L. Boasson and M. Nivat. Adherences of languages. *Journal of Computer and System Sciences*, 20:285–309, 1980.

J.E.R. Cury, B.A. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Trans. Autom. Control*, 43:564–568, 1998.

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

R. Kumar, V. Garg, and S.I. Marcus. On supervisory control of sequential behaviors. *IEEE Trans. Autom. Control*, 37: 1978–1985, 1992.

libFAUDES. Software library for discrete event systems. 2006–2011. URL `www.rt.eei.uni-erlangen.de/FGdes`.

Z. Manna and A. Pnueli. A hierarchy of temporal properties. *Proc. 9th ACM Symposium on Principles of Distributed Computing*, pages 377–408, 1990.

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

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

P.J. Ramadge. Some tractable supervisory control problems for discrete-event systems modeled by Büchi automata. *IEEE Trans. Autom. Control*, 34:10–19, 1989.

P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event systems. *SIAM J. Control and Optimization*, 25:206–230, 1987.

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

J.G. Thistle and W.M. Wonham. Supervision of infinite behavior of finite automata. *SIAM J. Control and Optimization*, 32: 1098–113, 1994a.

J.G. Thistle and W.M. Wonham. Control of infinite behavior of finite automata. *SIAM J. Control and Optimization*, 32: 1075–1097, 1994b.

J.C. Willems. Paradigms and puzzles in the theory of dynamic systems. *IEEE Trans. Autom. Control*, 36:258–294, 1991.

Th. Wittmann. Reglerentwurf für nicht-vollständige Systeme mit Ein- und Ausgängen. *Diplomarbeit, Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg*, 2010.