MODULAR SUPERVISORY CONTROL OF A CLASS OF HYBRID SYSTEMS IN A BEHAVIOURAL FRAMEWORK

T. Moor*, J.M. Davoren*, J. Raisch†

* Research School of Information Sciences and Engineering, Australian National University, Canberra
e-mail: thomas.moor@anu.edu.au, j.m.davoren@anu.edu.au
fax: +61 2 6125 8660, +61 2 6125 8651

† Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke Universität Magdeburg, FR Germany
e-mail: raisch@mpi-magdeburg.mpg.de
fax: +49 391 67-11191

Keywords: behavioural approach, modular control, hybrid systems, discrete approximation, supervisory control.

Abstract

This contribution investigates the discrete control of continuous or hybrid systems within the framework of behavioural systems theory. We address a problem of modularity, extending our recent work on approximation-based supervisory controller synthesis. More specifically, we identify conditions under which two discrete supervisors, each enforcing a particular specification, will have an admissible parallel composition that enforces both specifications simultaneously. While the main result corresponds to known facts from discrete event systems (DES) theory, it is our specific notion of inputs and outputs that enables the transfer of this result to a general class of hybrid systems.

1 Introduction

In a recent paper [3], we discussed the synthesis of supervisory control for hybrid systems with discrete external signals. The approach described in [3] is based on the notion of *l-complete approximation*, and is entirely set within the framework of Willems' behavioural systems theory. We extend this approach by investigating the problem of *modular control*. More specifically, we provide conditions that allow two supervisors, each enforcing a particular specification, to be combined to enforce both specifications simultaneously. The motivation for attempting modular control is twofold: (i) the synthesis of individual supervisors and their subsequent combination might be computationally less expensive than the direct synthesis of an overall controller; (ii) based on the concept of modular control, one may set up a "library" of supervisors, each geared towards a specific task for a given plant; depending on the particular application situation (corresponding to a certain combination of tasks), the appropriate controllers can then be simply retrieved from the library and run in parallel to solve the problem at hand.

The paper is organised as follows. In Section 2, we collect some basic facts from Willems' behavioural systems theory and from automata theory. In Section 3, we give a short overview of approximation based supervisory control for hybrid systems. In Section 4, this is extended to modular control. An illustrative example is given in Section 5.

2 Hybrid systems in a behavioural framework

The purpose of this section is to collect some basic definitions from Willems' behavioural systems theory and from automata theory, and to provide a link to the class of hybrid systems that we will consider subsequently.

Definition 1. (See [9], Def. II.1) A *dynamical system* Σ is a triple $(T,\ W,\ \mathfrak{B})$, with $T \subseteq \mathbb{R}$ the *time axis*, W the *signal space*, and $\mathfrak{B} \subseteq W^T := \{f \mid f : T \to W\}$ the *behaviour*. □

The behaviour is viewed as the set of all trajectories which are compatible with the phenomena modelled by the system: trajectories $w \notin \mathfrak{B}$ cannot occur. An overview of the behavioural framework is given in [9] and [8], including definitions of fundamental properties like time-invariance, linearity, controllability, completeness, inputs and outputs or the state axiom in terms of behaviours. Within this paper, we consider systems with discrete time axis $T = \mathbb{N}_0$ and use a specific notion of input and output signals[1].

Definition 2. (see [9], Def. VIII.1 and VIII.4) The system $\Sigma = (\mathbb{N}_0,\ W,\ \mathfrak{B})$, $W = U \times Y$, is said to be an *I/- system* if:

(i) the input is free, i.e. $\mathcal{P}_U\ \mathfrak{B} = U^{\mathbb{N}_0}$ (where $\mathcal{P}_U$ denotes the projection onto $U^{\mathbb{N}_0}$);

(ii) the output does not anticipate the input, i.e. for all $t \in \mathbb{N}_0,\ (\tilde{u},\ \tilde{y}),\ (\hat{u},\ \hat{y}) \in \mathfrak{B}$ the following implication holds:

$$\tilde{u}\big|_{[0,t]} = \hat{u}\big|_{[0,t]} \implies \exists\, y \in Y^{\mathbb{N}_0} :\ y\big|_{[0,t]} = \tilde{y}\big|_{[0,t]},\ \text{and}\ (\hat{u},\ y) \in \mathfrak{B}\,. \quad (1)$$

Here, $w|_{[t_1,t_2]},\ t_1 \le t_2$, is the restriction of a map $w : \mathbb{N}_0 \to W$ to the domain $[t_1, t_2] \cap \mathbb{N}_0$. □

Discrete-time systems can be realized by state machines, as formalised below.

Definition 3. Let the sets $W,\ X,\ X_0 \subseteq X,\ \delta \subseteq X \times W \times$

[1] $\mathbb{N}$: natural numbers without 0; $\mathbb{N}_0$: natural numbers including 0.

X denote the *external signal space*, the *state space*, the *set of initial conditions* and the *next state relation*, respectively. The tuple $P = (X,\ W,\ \delta,\ X_0)$ is called a *state machine*. If $|W| \in \mathbb{N}$ and $|X| \in \mathbb{N}$ (both sets are finite), P is said to be a *finite state machine*. The behaviour

$$\mathfrak{B}_s := \{(w, x) | \\ \forall\, t \in \mathbb{N}_0 \colon\ (x(t),\ w(t),\ x(t+1)) \in \delta \text{ and } x(0) \in X_0\}$$

is referred to as the induced *full behaviour*, and $\Sigma_s := (\mathbb{N}_0,\ W \times X,\ \mathfrak{B}_s)$ as the induced state space system. The *external behaviour* $\mathfrak{B}_{ex}$ of Σ_s is defined to be the projection of $\mathfrak{B}_s$ onto $W^{\mathbb{N}_0}$, i. e. $\mathfrak{B}_{ex} := \mathcal{P}_W \mathfrak{B}_s := \{w | \exists\, x : (w, x) \in \mathfrak{B}_s\}$. Conversely, a state machine P' with induced external behaviour $\mathfrak{B}'$ is said to be a *realization* of the system $\Sigma' = (\mathbb{N}_0,\ W,\ \mathfrak{B}')$. This is denoted by $\Sigma' \cong P'$. □

We now introduce some basic terminology related to state machines:

Definition 4. Consider state machines $P_a = (A,\ W,\ \alpha,\ A_0)$ and $P_b = (B,\ W,\ \beta,\ B_0)$. A state $a_1 \in A$ is said to be *reachable* if there exists a state $a_0 \in A_0$ and a sequence of transitions (elements in the next state relation) from α connecting a_0 with a_1. The state machine P_a is said to be *reachable* if every state $a_1 \in A$ is reachable. The state machine P_a is called *non-blocking* if for every reachable state $a \in A$, there exists $w \in W$ and $a' \in A$ such that $(a,\ w,\ a') \in \alpha$. Referring to $W = U \times Y$, the state machine P_a is said to be an *I/S/- machine*, if for every reachable $a \in A$, $\mu \in U$, there exist $\nu \in Y$, $a' \in A$ such that $(a,\ (\mu,\ \nu),\ a') \in \alpha$. The *parallel composition* of P_a and P_b is defined by $P_a \parallel P_b := (A \times B,\ W,\ \lambda,\ A_0 \times B_0)$, where $((a, b),\ w,\ (a', b')) \in \lambda$ if and only if $(a,\ w,\ a') \in \alpha$ and $(b,\ w,\ b') \in \beta$. □

A typical strategy within the behavioural approach is to work out the relationship between properties defined in terms of behaviours and corresponding properties of realizations: e.g. if P is an I/S/- machine, the induced system Σ is an I/- system; see [3], Proposition 24.

Definition 5. (See [8], Section 2.2.1) Let $\mathfrak{B}_s$ be the full behaviour induced by the state machine $P = (X,\ W,\ \delta,\ X_0)$. Then P is said to be *past-induced* if $t \in \mathbb{N}_0$, $(w',\ x')$, $(w'',\ x'') \in \mathfrak{B}_s$ and $w'|_{[0,t)} = w''|_{[0,t)}$ implies $x'(t) = x''(t)$. □

A past-induced state machine is "instantaneously state observable": for every $t \in \mathbb{N}_0$, we can figure out $x(t)$ by only investigating the past $w|_{[0,t)}$ of the external signal. Thus past-inducedness is a crucial property for control related tasks. In fact, past-induced realizations of the plant model are the setting for Ramadge and Wonham's DES supervisory control theory.

We address a class of hybrid systems that are realizable by $I/S/-$ machines and that are further characterized by the fact that their external signal space is finite (i.e. $|W| \in \mathbb{N}$), while their state set X is a product of $\mathbb{R}^n$ and a finite set D. These assumptions on the stucture of the plant are rather weak and allow us to cover characteristic features of various more detailed hybrid models; e.g. hybrid automata [1]. In Section 5 we give an example within the class of switched flow systems. Here, the discrete time axis is interpreted as "logic time", and refers to the enumeration of the occurrence of events which in turn are defined as certain continuous variables crossing certain threshold values.

3 Approximation based supervisory control

The problem of supervisory controller synthesis as studied in DES theory (e.g. [5, 10]) can be stated within the behavioural framework: *given a plant model* Σ_p *and a specification* Σ_{spec} *of the acceptable closed-loop behaviour, construct a supervisor* Σ_{sup} *such that the interconnection of* Σ_p *and* Σ_{sup} *exhibits a behaviour not exceeding the one specified by* Σ_{spec}. Rather than considering a disjoint union of controllable and uncontrollable events (as in [5, 10]), we set up the external signal space by a product composition $W = U \times Y$ and assume the plant to be an I/- system. This turns out to be crucial for the synthesis of supervisors based on a finite automaton approximation of the plant. In the following, we collect the main results of our approach. A detailed discussion can be found in [3].

First, we need to examine system interconnection. As usual, the intention is the synchronization of the external variable(s); hence system interconnection —in principle— corresponds to the intersection of the external behaviours. However, two conditions apply to this scenario of interconnected systems, both motivated from an application point of view.

(i) The synchronization shall be performed "locally on the time axis", i.e. at any instance of time and independent of the past evolution, it shall be clear on which value the two systems can agree without "getting stuck" in the future. This demand corresponds to the notion of *non-conflicting* languages in DES theory.

(ii) The supervisor shall take its effect on the plant via some actuator and in turn read back measurements by some sensor referring to the input and the output component of the external signal respectively. Thus, the supervisor must not directly affect the output component of the external signal in order to be *implementable*.

These two conditions can both be stated in terms of behaviours:

Definition 6. Consider two systems $\Sigma_a = (\mathbb{N}_0,\ W,\ \mathfrak{B}_a)$ and $\Sigma_b = (\mathbb{N}_0,\ W,\ \mathfrak{B}_b)$ over the same signal-space W.

(i) Σ_a and Σ_b are said to be *non-conflicting* (w.r.t. each other) if the following holds for all $t \in \mathbb{N}_0$:

$$\mathfrak{B}_a\big|_{[0,t]} \cap \mathfrak{B}_b\big|_{[0,t]} = (\mathfrak{B}_a \cap \mathfrak{B}_b)\big|_{[0,t]}\,. \tag{2}$$

(ii) Given a decomposition $W = U \times Y$ into inputs and outputs of Σ_a, the system Σ_b is said to be *implementable* w.r.t Σ_a if for all $t \in \mathbb{N}_0$, $\bar{w}|_{[0,t]} \in \mathfrak{B}_a|_{[0,t]} \cap \mathfrak{B}_b|_{[0,t]}$ and $\hat{w}|_{[0,t]} \in \mathfrak{B}_a|_{[0,t]}$ the following holds:

$$\hat{w}|_{[0,t)} = \bar{w}|_{[0,t)} \text{ and } \mathcal{P}_U\, \hat{w}(t) = \mathcal{P}_U\, \bar{w}(t) \implies \hat{w}|_{[0,t]} \in \mathfrak{B}_{\mathrm{b}}|_{[0,t]}\,. \quad (3)$$

A supervisor $\Sigma_{\mathrm{sup}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{sup}})$ is said to be *admissible* w.r.t. a plant $\Sigma_{\mathrm{p}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{p}})$ if Σ_{p} and Σ_{sup} are non-conflicting and Σ_{sup} is implementable w.r.t. Σ_{p}. □

If an admissible supervisor Σ_{sup} is interconnected with the plant Σ_{p}, the closed-loop is given by the system $\Sigma_{\mathrm{cl}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{cl}})$, where $\mathfrak{B}_{\mathrm{cl}} = \mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{sup}}$. It can bee seen that if Σ_{sup} is admissible w.r.t. Σ_{p}, then so is the closed-loop Σ_{cl}; i.e. Σ_{cl} may also serve as a supervisor.

Given a plant Σ_{p} and a specification $\Sigma_{\mathrm{spec}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{spec}})$ of acceptable closed-loop trajectories, an admissible supervisors Σ_{sup} which achieves $\mathfrak{B}_{\mathrm{cl}} \subseteq \mathfrak{B}_{\mathrm{spec}}$ is said to be a *solution* to the supervisory controller synthesis problem. Note that formally there always exists the trivial solution $\Sigma_{\emptyset} = (\mathbb{N}_0,\ W,\ \emptyset)$. The trivial solution leads to an empty closed-loop behaviour and thus is not desirable. In analogy with the scenario studied in DES theory, all the demands we put on a solution Σ_{sup} are seen to be retained under arbitrary union of supervisor behaviours; see [3], Proposition 17. Hence, when Σ_{p} and Σ_{spec} are given, the union $\mathfrak{B}^{+}_{\mathrm{sup}}$ over all solution behaviours $\mathfrak{B}_{\mathrm{sup}}$ itself determines a solution, namely $\Sigma^{+}_{\mathrm{sup}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}^{+}_{\mathrm{sup}})$. As all solution behaviours are contained in $\mathfrak{B}^{+}_{\mathrm{sup}}$, the system $\Sigma^{+}_{\mathrm{sup}}$ is referred to as the *least restrictive* solution to the control problem. It is also observed that the least restrictive supervisor, when connected to the plant, gives rise to the least restrictive closed-loop behaviour. Obviously, a non-trivial closed-loop exists if and only if the least restrictive closed-loop behaviour is non-empty. A synthesis procedure may focus its search for admissible supervisors on behaviours contained in $\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{spec}}$ without risk of missing non-trivial solutions.

The following definition intends to capture the notion of admissibility on the realization level.

Definition 7. Let $P = (X,\ W,\ \delta,\ X_0)$ and $P_{\mathrm{spec}} = (X_{\mathrm{spec}},\ W,\ \delta_{\mathrm{spec}},\ X_{\mathrm{spec}_0})$ be state machines. Let $P_{\|} := (Q,\ W,\ \lambda,\ Q_0) := P \parallel P_{\mathrm{spec}}$. The transitions $(\xi,\ \omega,\ \xi') \in \delta$ and $(\tilde{\xi},\ \tilde{\omega},\ \tilde{\xi}') \in \delta$ are called *partners*, if $\xi = \tilde{\xi}$ and $\mathcal{P}_U\, \omega = \mathcal{P}_U\, \tilde{\omega}$. Let $\tilde{P}_{\|} = (Q,\ W,\ \tilde{\lambda},\ \tilde{Q}_0)$ be a state machine such that

(i) $\tilde{\lambda} \subseteq \lambda$, $\tilde{Q}_0 \subseteq Q_0$, and

(ii) a transition $((\xi,\ \xi_{\mathrm{spec}}),\ \omega,\ (\xi',\ \xi'_{\mathrm{spec}})) \in \lambda$ can only be an element in $\tilde{\lambda}$ if for every partner $(\xi,\ \tilde{\omega},\ \tilde{\xi}')$ of $(\xi,\ \omega,\ \xi')$ there exists a $\tilde{\xi}'_{\mathrm{spec}} \in X_{\mathrm{spec}}$ such that the transition $((\xi,\ \xi_{\mathrm{spec}}),\ \tilde{\omega},\ (\tilde{\xi}',\ \tilde{\xi}'_{\mathrm{spec}}))$ is in $\tilde{\lambda}$.

Then, $\tilde{P}_{\|}$ is called a *substructure* of $P \parallel P_{\mathrm{spec}}$ w.r.t. P. □

Indeed, any system Σ_{sup} realized by a non-blocking substructure of $P \parallel P_{\mathrm{spec}}$ w.r.t. P is an admissible supervisor w.r.t. $\Sigma_{\mathrm{p}} \cong P$ enforcing the specifications $\Sigma_{\mathrm{spec}} \cong P_{\mathrm{spec}}$; see [3], Proposition 17. Moreover, if P and P_{spec} are non-blocking and past-induced, the least restrictive closed-loop system can be realized by a non-blocking substructure; see [3], Theorem 21. Thus, for *finite* past-induced realizations P and P_{spec}, one may first remove transitions leading into blocking states, then form the parallel composition and finally, in the parallel composition, remove all partners of transitions leading into blocking states. The outcome of this procedure is a state machine realizing the least restrictive closed-loop system Σ^{+}_{cl} and thus solving the synthesis problem on realization level.

If the plant is hybrid, however, a finite realization will not in general exist. On the other hand, if the external signal space is a finite set, the property of *l-completeness*, $l \in \mathbb{N}$, (see [9]) serves as a sufficient condition for the existence of a finite past-induced realization. Hence it is tempting to conservatively approximate the plant behaviour $\mathfrak{B}_{\mathrm{p}}$ by some l-complete behaviour $\mathfrak{B}_l$ and then to consider the synthesis problem for $\Sigma_l = (\mathbb{N}_0,\ W,\ \mathfrak{B}_l)$. The crucial question is then whether a solution of the control problem for the approximation also solves the problem for the plant Σ_{p} (i.e. whether the resulting supervisor is admissable w.r.t. the plant and enforces the specifications for Σ_{p}). The following theorem (see [3], Theorem 25) gives an affirmative answer:

Theorem 8. Let $\Sigma_{\mathrm{ca}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{ca}})$ be a conservative approximation of $\Sigma_{\mathrm{p}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{p}})$, i.e. $\mathfrak{B} \subseteq \mathfrak{B}_{\mathrm{ca}}$; let $\Sigma_{\mathrm{sup}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}_{\mathrm{sup}})$ be a complete admissible supervisor w.r.t. Σ_{ca}. If Σ_{p} is a complete I/- system, then Σ_{sup} is an admissible supervisor w.r.t. Σ_{p}. If the closed-loop behaviour $\mathfrak{B}_{\mathrm{ca}} \cap \mathfrak{B}_{\mathrm{sup}}$ is nonempty, so is $\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{sup}}$. □

Note that both the specifications we consider and the supervisor our procedure comes up with are finite state-machines and thus are known to induce a complete external behaviour. The completeness requirement for Σ_{p} can also be dropped if the plant is realized by an I/S/- machine.

In [3], we provide a general method for the construction of a past-induced non-blocking finite state machine P_l that realizes the strongest l-complete approximation Σ_l of Σ_{p}. For the case of linear time-invariant continuous-time systems with discrete external variables, a detailed computational procedure introducing a second approximation step has been derived in [2]. This —in principle— allows approximation based synthesis of discrete supervisory control for a rich class of hybrid plant models. It is obvious, however, that on the (DES) approximation level we encounter the "curse of dimensionality". This is one of the reasons for our exploration of modular extensions of our approach.

4 Modular control

Setting up an overall supervisor by combining a number of individual supervisors is referred to as *modular supervisory control*. There are two potential benefits from modular supervisors. First, it may turn out that the synthesis of individual supervisors and their combination is computationally cheaper than the direct synthesis of an overall supervisor. Second, given a plant, one may set up a library of supervisors which can be

combined in order to suit various applications for that plant. In the field of DES theory, this topic has been studied extensively, e.g. [6, 7, 10]. As with our "non-modular" version of supervisory control, it is expected that basic principles carry over to the behavioural framework and thus can be employed to establish modular controller synthesis for hybrid systems.

Given a plant Σ_p, we condider the situation in which the problem of supervisory control has been solved for two specifications $\Sigma_{\mathrm{spec,a}}$ and $\Sigma_{\mathrm{spec,b}}$ individually. That is, two supervisors $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ have been established, both admissible w.r.t. the plant and —when connected individually— enforcing the desired specifications $\Sigma_{\mathrm{spec,a}}$ and $\Sigma_{\mathrm{spec,b}}$, respectively. Here, the sensible question to ask is under which circumstances and how these two supervisors can be combined into an overall supervisor Σ_sup such that both specifications are enforced simultaneously, i.e. the closed-loop behaviour $\mathfrak{B}_\mathrm{cl} = \mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_\mathrm{sup}$ must be a subset of the intersection $\mathfrak{B}_\mathrm{spec} \coloneqq \mathfrak{B}_{\mathrm{spec,a}} \cap \mathfrak{B}_{\mathrm{spec,b}}$. A natural starting point here is to run both supervisors $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ in parallel. Thus $\mathfrak{B}_\mathrm{sup} = \mathfrak{B}_{\mathrm{sup,a}} \cap \mathfrak{B}_{\mathrm{sup,b}}$ is our candidate for the behaviour of the overall supervisor. Trivially, this approach leads to $\mathfrak{B}_\mathrm{cl} \subseteq \mathfrak{B}_\mathrm{spec}$. However, while the property of admissibility for supervisors is preserved under union, the corresponding statement does not hold true for intersections. Thus, we ask for a criterion which guarantees our candidate Σ_sup to be admissible w.r.t. the plant Σ_p. Furthermore, as on the realization level we intend to run $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ in parallel, we require that they do not conflict as long as the trajectory evolves within the plant behaviour. Formally, we require

$$\begin{aligned} & \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,a}}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,b}}\big|_{[0,t]} \\ = \; & \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap (\mathfrak{B}_{\mathrm{sup,a}} \cap \mathfrak{B}_{\mathrm{sup,b}})\big|_{[0,t]} \end{aligned}$$

for all $t \in \mathbb{N}_0$ and refer to this property as $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ being *non-conflicting relative to* Σ_p.

Proposition 9. Let both supervisors $\Sigma_{\mathrm{sup},i} = (\mathbb{N}_0, W, \mathfrak{B}_{\mathrm{sup},i})$ be non-conflicting w.r.t. the plant $\Sigma_\mathrm{p} = (\mathbb{N}_0, W, \mathfrak{B}_\mathrm{p})$, and denote the individual closed-loop systems by $\Sigma_{\mathrm{cl},i} = (\mathbb{N}_0, W, \mathfrak{B}_{\mathrm{cl},i})$, $\mathfrak{B}_{\mathrm{cl},i} = \mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_{\mathrm{sup},i}$, where $i \in \{\mathrm{a}, \mathrm{b}\}$. Then for the combined supervisor $\Sigma_\mathrm{sup} = (\mathbb{N}_0, W, \mathfrak{B}_\mathrm{sup})$, $\mathfrak{B}_\mathrm{sup} = \mathfrak{B}_{\mathrm{sup,a}} \cap \mathfrak{B}_{\mathrm{sup,b}}$, the following are equivalent

(i) $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ are non-conflicting relative to Σ_p, and Σ_sup and Σ_p are non-conflicting;

(ii) $\Sigma_{\mathrm{cl,a}}$ and $\Sigma_{\mathrm{cl,b}}$ are non-conflicting.

Proof. First, we assume (i) to hold true and establish (ii) by the following observation:

$$\begin{aligned} & (\mathfrak{B}_{\mathrm{cl,a}} \cap \mathfrak{B}_{\mathrm{cl,b}})\big|_{[0,t]} \\ = \; & (\mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_\mathrm{sup})\big|_{[0,t]} \quad = \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap \mathfrak{B}_\mathrm{sup}\big|_{[0,t]} \\ = \; & \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,a}}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,b}}\big|_{[0,t]} \\ = \; & \mathfrak{B}_{\mathrm{cl,a}}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{cl,b}}\big|_{[0,t]} . \end{aligned}$$

This proves "(i)⇒(ii)". Conversely, we now assume (ii) to hold. Note that

$$\mathfrak{B}_\mathrm{a}\big|_{[0,t]} \cap \mathfrak{B}_\mathrm{b}\big|_{[0,t]} \supseteq (\mathfrak{B}_\mathrm{a} \cap \mathfrak{B}_\mathrm{b})\big|_{[0,t]} \tag{4}$$

holds true for any two behaviours $\mathfrak{B}_\mathrm{a}$ and $\mathfrak{B}_\mathrm{b}$ over the same signal-space, and thus observe

$$\begin{aligned} & \mathfrak{B}_\mathrm{sup}\big|_{[0,t]} \cap \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \\ \subseteq \; & \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,a}}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,b}}\big|_{[0,t]} \\ = \; & (\mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_{\mathrm{sup,a}})\big|_{[0,t]} \cap (\mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_{\mathrm{sup,b}})\big|_{[0,t]} \\ = \; & (\mathfrak{B}_{\mathrm{cl,a}} \cap \mathfrak{B}_{\mathrm{cl,b}})\big|_{[0,t]} \quad = (\mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_\mathrm{sup})\big|_{[0,t]} . \end{aligned}$$

Again by Eq. (4), this implies

$$(\mathfrak{B}_\mathrm{sup} \cap \mathfrak{B}_\mathrm{p})\big|_{[0,t]} = \mathfrak{B}_\mathrm{sup}\big|_{[0,t]} \cap \mathfrak{B}_\mathrm{p}\big|_{[0,t]} , \tag{5}$$

and Σ_sup is non-conflicting w.r.t. Σ_p. In particular, the "$\subseteq$" relation in the preceeding consideration can only hold true by equality. Hence, we conclude

$$\mathfrak{B}_\mathrm{sup}\big|_{[0,t]} \cap \mathfrak{B}_\mathrm{p}\big|_{[0,t]} = \mathfrak{B}_\mathrm{p}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,a}}\big|_{[0,t]} \cap \mathfrak{B}_{\mathrm{sup,b}}\big|_{[0,t]} ,$$

and $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ are non-conflicting relative to Σ_p. This completes the proof of "(ii)⇒(i)". □

By the above proposition, a necessary and sufficient criterion for the desired non-conflicting properties is given in terms of the individual closed-loop behaviours. In order to guarantee that the combined supervisor is admissible w.r.t. the plant, we must further examine the issue of implementability. It turns out that the non-conflicting property of individual closed-loop behaviours can again be used as a criterion:

Theorem 10. Let the plant $\Sigma_\mathrm{p} = (\mathbb{N}_0, W, \mathfrak{B}_\mathrm{p})$, $W = U \times Y$, be an I/- system. Let both supervisors $\Sigma_{\mathrm{sup},i} = (\mathbb{N}_0, W, \mathfrak{B}_{\mathrm{sup},i})$ be admissible w.r.t. the plant $\Sigma_\mathrm{p} = (\mathbb{N}_0, W, \mathfrak{B}_\mathrm{p})$, and denote the individual closed-loop systems by $\Sigma_{\mathrm{cl},i} = (\mathbb{N}_0, W, \mathfrak{B}_{\mathrm{cl},i})$, $\mathfrak{B}_{\mathrm{cl},i} = \mathfrak{B}_\mathrm{p} \cap \mathfrak{B}_{\mathrm{sup},i}$, where $i \in \{\mathrm{a}, \mathrm{b}\}$. If $\Sigma_{\mathrm{cl,a}}$ and $\Sigma_{\mathrm{cl,b}}$ are non-conflicting, then the supervisor $\Sigma_\mathrm{sup} = (\mathbb{N}_0, W, \mathfrak{B}_\mathrm{sup})$, $\mathfrak{B}_\mathrm{sup} = \mathfrak{B}_{\mathrm{sup,a}} \cap \mathfrak{B}_{\mathrm{sup,b}}$, is admissible w.r.t. the plant Σ_p.

Proof. From Proposition 9 it is known that Σ_sup is non-conflicting w.r.t. Σ_p. In order to show that it is also implementable, pick any $t \in \mathbb{N}_0$, $\bar{w}|_{[0,t]} \in \mathfrak{B}_\mathrm{p}|_{[0,t]} \cap \mathfrak{B}_\mathrm{sup}|_{[0,t]}$ and $\hat{w}|_{[0,t]} \in \mathfrak{B}_\mathrm{p}|_{[0,t]}$ such that $\hat{w}|_{[0,t)} = \bar{w}|_{[0,t)}$ and $\mathcal{P}_U\, \hat{w}(t) = \mathcal{P}_U\, \bar{w}(t)$. We need to establish $\hat{w}|_{[0,t]} \in \mathfrak{B}_\mathrm{sup}|_{[0,t]}$. Since $\mathfrak{B}_\mathrm{sup} = \mathfrak{B}_{\mathrm{sup,a}} \cap \mathfrak{B}_{\mathrm{sup,b}}$ and both $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ are implementable w.r.t. Σ_p, it is observed that $\hat{w}|_{[0,t]} \in \mathfrak{B}_{\mathrm{sup,a}}|_{[0,t]}$ and $\hat{w}|_{[0,t]} \in \mathfrak{B}_{\mathrm{sup,b}}|_{[0,t]}$. Hence, $\hat{w}|_{[0,t]} \in \mathfrak{B}_{\mathrm{cl,a}}|_{[0,t]} \cap \mathfrak{B}_{\mathrm{cl,b}}|_{[0,t]}$. As $\Sigma_{\mathrm{cl,a}}$ and $\Sigma_{\mathrm{cl,b}}$ are non-conflicting, this implies $\hat{w}|_{[0,t]} \in (\mathfrak{B}_{\mathrm{cl,a}} \cap \mathfrak{B}_{\mathrm{cl,b}})|_{[0,t]}$ and thus $\hat{w}|_{[0,t]} \in \mathfrak{B}_\mathrm{sup}|_{[0,t]}$. □

From the behavioural point of view, we may run the individual supervisors $\Sigma_{\mathrm{sup,a}}$ and $\Sigma_{\mathrm{sup,b}}$ in parallel on the plant Σ_p and by this achieve both specifications $\Sigma_{\mathrm{spec,a}}$ and $\Sigma_{\mathrm{spec,b}}$ simultaneously if and only if the individual closed-loops $\Sigma_{\mathrm{cl,a}}$ and $\Sigma_{\mathrm{cl,b}}$ are non-conflicting. Furthermore, any admissable supervisor Σ_sup which enforces both specifications $\Sigma_{\mathrm{spec,a}}$ and $\Sigma_{\mathrm{spec,b}}$ simultaneuosly must exhibit a behaviour $\mathfrak{B}_\mathrm{sup}$ not exceeding the induvidual least restrictive supervisor behaviours

$\mathfrak{B}^+_{\text{sup,a}}$ and $\mathfrak{B}^+_{\text{sup,b}}$. Thus, if the combined supervisor $\Sigma^+_{\text{sup}} = (\mathbb{N}_0,\ W,\ \mathfrak{B}^+_{\text{sup}})$, $\mathfrak{B}^+_{\text{sup}} = \mathfrak{B}^+_{\text{sup,a}} \cap \mathfrak{B}^+_{\text{sup,b}}$ is admissible at all, Σ^+_{sup} is seen to be the least restrictive admissible supervisor enforcing both specifications simultaneously.

We now work out the consequences of the above result on the realization level. First, it is assumed that Σ_{p}, $\Sigma_{\text{spec,a}}$ and $\Sigma_{\text{spec,b}}$ are realized by past-induced non-blocking finite state machines P, $P_{\text{spec,a}}$ and $P_{\text{spec,b}}$ respectively. Our synthesis procedure then establishes non-blocking past-induced finite realizations $P^+_{\text{cl,a}}$ and $P^+_{\text{cl,b}}$ of the supremal closed-loop systems $\Sigma^+_{\text{cl,a}}$ and $\Sigma^+_{\text{cl,b}}$ which serve as individual supervisors. In the particular case of non-blocking past-induced realizations, the induced external behaviours are non-conflicting if and only if the parallel composition is non-blocking. This can be checked by examining all reachable states in $P^+_{\text{cl,a}} \parallel P^+_{\text{cl,b}}$.

For the case the plant Σ_{p} is a hybrid system, we apply the results of [3] to synthesise individual supervisors for each of the specifications based on a conservative approximation of the plant, as discussed in Section 3. One then ends up in a situation similar to the finite state case, but with the fundamental difference that the established realizations $P_{\text{cl,a}}$ and $P_{\text{cl,b}}$ —which can be employed as supervisors— are based on a finite approximation rather than on the actual hybrid plant. Still, Proposition 9 and Theorem 10 hold true and in principle one could develop a procedure to check for non-conflictingness of the actual closed-loop behaviours. However, as our candidate overall supervisor is realized by $P_{\text{cl,a}} \parallel P_{\text{cl,b}}$, the parallel composition is required to be non-blocking anyway. If this is indeed the case, Theorem 10 guarantees $P_{\text{cl,a}} \parallel P_{\text{cl,b}}$ to realize a supervisor which is admissible w.r.t. the approximation. Then Theorem 8 in turn guarantees that our candidate overall supervisor is admissible w.r.t. the actual plant.

In terms of realizations —whether they be finite or hybrid— the results sum up to the simple formula: if the parallel composition $P_{\text{cl,a}} \parallel P_{\text{cl,b}}$ is non-blocking, it can be employed as a supervisor enforcing both specifications simultaneuosly.

5 Example

We consider a thermal switched-server system consisting of two plates and a radiator, similar to the one described in [2]. The radiator can either be switched off or on, heating a single plate depending on its position. A switching strategy has to be implemented by a supervisor in order to (a) keep the temperatures of all plates in a specified range while (b) maintaining an upper bound on the switching frequency. In [2] we solve this control problem by directly applying the methods developed in [3], and we also give a detailed account of computational aspects. Here, we first treat both specifications (a) and (b) separately and then combine the resulting supervisors such that both specifications are enforced simultaneously.

In setting up a plant model, we refer to the following parameters: the radiator and the environment temperatures $\beta_r \in \mathbb{R}$ and $\beta_e \in \mathbb{R}$, respectively; the corresponding normalized heat transfer coefficients $\alpha_r,\ \alpha_e \in \mathbb{R}^+$; the specified range of allowed temperatures $[\beta_-,\ \beta_+] \subset \mathbb{R}$; it is assumed that the initial temperatures lie within $(\beta_0,\ \beta_+) \subset \mathbb{R}$. The temperature $x_i(\,\cdot\,)$ of plate $i \in \{1,\ 2\}$ is modelled either by equation (6) when it is heated or by equation (7) when it is not heated:

$$\dot{x}_i(t) = \alpha_r\ (\beta_r - x_i(t)) + \alpha_e\ (\beta_e - x_i(t))\ , \tag{6}$$

$$\dot{x}_i(t) = 2\ \alpha_e\ (\beta_e - x_i(t))\ . \tag{7}$$

Observe that $x_i(t) \equiv \beta_m \coloneqq (\alpha_r\ \beta_r + \alpha_e\ \beta_e)/(\alpha_r + \alpha_e)$ is a stable equilibrium for a heated plate, and $x_i(t) \equiv \beta_e$ for a non-heated plate. We assume parameter values such that $\beta_e < \beta_- < \beta_0 < \beta_+ < \beta_m < \beta_r$ holds.

Whenever the temperature of plate $\nu_{plt} \in \{1,\ 2\}$ reaches the threshold $\nu_{val} \in \{\beta_-,\ \beta_0,\ \beta_+\}$, the output signal $(\nu_{plt},\ \nu_{val}) \in Y = \{1,\ 2\} \times \{\beta_-,\ \beta_0,\ \beta_+\}$ is generated. In response, the supervisor may disable certain discrete input signals from $\mu \in U = \{1,\ 2,\ 3\}$, where $\mu = 3$ is interpreted as "radiator off", while $\mu < 3$ is interpreted as "radiator positioned at plate μ". If more than one input signal is enabled, selection is instantaneous — either at random or by some higher level control device.

The external plant behaviour $\mathfrak{B}_{\text{p}} \subseteq (U \times Y)^{\mathbb{N}_0}$ is then defined to be the set all those sequences of input and output events that can occur according to our model; for a formal definition of the external behaviour of switched continuous systems see e.g. [2, 4]. Note that while the external behaviour is discrete in both time-axis and signal space, the plant behaviour $\mathfrak{B}_{\text{p}}$ is determined by the continuous dynamics in continuous time as given by the differential equations (6) and (7). In general, hybrid systems are known to exhibit complex dynamics, and the analysis of a hybrid system can turn to out to be highly non-trivial.

In order to formalise our control problem, we state both specifications in terms of the external events:

(a) If plate ν_{plt} triggers the threshold $\nu_{val} = \beta_+$, the next input must not heat that plate. If plate ν_{plt} triggers the threshold $\nu_{val} = \beta_-$, the next input must heat that plate.

(b) No reheating process must be started for a plate at a temperature above β_0. Once the reheating process of a plate has been started, it has to be continued until the plate temperature reaches β_+.

Clearly, a consequence of enforcing specification (a) is that the temperatures of the plates are kept within $[\beta_-, \beta_+]$. Specification (b) requires that any reheating process goes at least from $[\beta_0, \beta_+]$. By Eq. (6), such a reheating process is seen to have a minimal duration of

$$T = \frac{1}{\alpha_r + \alpha_e}\ \ln \frac{\beta_m - \beta_0}{\beta_m - \beta_+}\ . \tag{8}$$

Hence, in fulfilling (b) a supervisor enforces an upper bound on the frequency of radiator re-allocation.

Treating both specifications (a) and (b) separately, we use the method proposed in [2] to construct two past-induced finite realizations $P_{\text{cl,a}}$ and $P_{\text{cl,b}}$ that enforce (a) and (b), respectively.

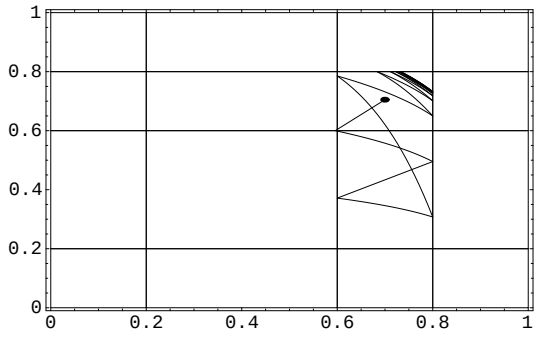

Figure 1: hybrid plant under supervision $P_{\mathrm{cl,a}}$

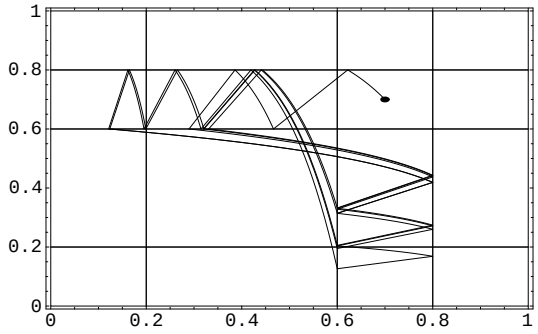

Figure 2: hybrid plant under supervision $P_{\mathrm{cl,b}}$

Fig. 1 and 2 show closed-loop simulations, where the parameter values are chosen as $\beta_e = 0$, $\beta_r = 1$, $\alpha_e = 0.1$, $\alpha_r = 1.0$, $\beta_+ = 0.8$, $\beta_0 = 0.6$, $\beta_- = 0.2$, and the synthesis is based on an l-complete approximation with $l = 4$. The simulation illustrates the capability of the individual supervisors to enforce their respective specifications. The figures also give evidence that $P_{\mathrm{cl,a}}$ does not enforce specification (b) and that $P_{\mathrm{cl,b}}$ does not enforce specification (a). However, the parallel composition $P_{\mathrm{cl,a}} \parallel P_{\mathrm{cl,b}}$ turns out to be non-blocking, and, as both supervisors are past-induced, this implies that the induced behaviours are non-conflicting. Thus $P_{\mathrm{cl,a}} \parallel P_{\mathrm{cl,b}}$ realises an admissible supervisor that enforces both (a) and (b) simultaneously, as illustrated by the closed-loop simulation in Fig. 3.

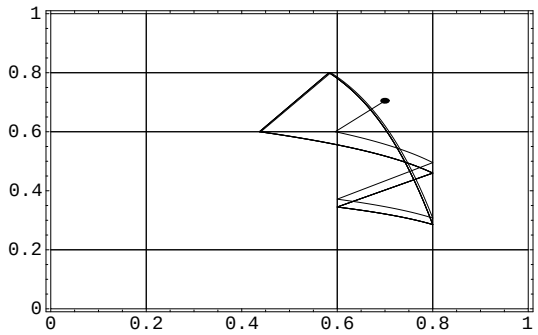

Figure 3: hybrid plant under supervision $P_{\mathrm{cl,a}} \parallel P_{\mathrm{cl,b}}$

6 Conclusions

A problem of modular supervisory control is considered within the framework of behavioural systems theory. We propose a condition under which two supervisors, each enforcing a particular specification, will have an admissible parallel composition that enforces both specifications simultaneously. While similar results have been known from DES theory, our notion of admissible supervisors refers to the definition of inputs and outputs in the behavioural style and in particular suits a general class of hybrid plant models. Thus the investigated principles of modular supervisory control can be seen as an extension to recent work on supervisory control of hybrid systems. Potential benefits gained by modularity are seen to be applicable to the synthesis of discrete controllers for hybrid systems: (i) the synthesis of individual supervisors and their combination may turn out to be computationally cheaper than the direct synthesis of an overall supervisor; (ii) one may set up a library of supervisors which can be combined in order to suit various applications for a specific plant.

Acknowledgements: Research partially supported by US Office of Naval Research, Grant N 00014-98-1-0535.

References

[1] Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G. 2000. Discrete Abstractions of Hybrid Systems. *Proceedings of the IEEE*, vol. 88:7, pp. 971–984.

[2] Franke, D., Moor, T., Raisch, J. 2000. Discrete supervisory control of switched linear systems. *at-Automatisierungstechnik*, vol. 48:9, pp. 461–467.

[3] Moor, T., Raisch, J. 1999. Supervisory control of hybrid systems within a behavioural framework. In Evans, R. J. and Savkin, A. V. (eds.) *Systems and Control Letters*, vol 38, no 3, pp. 157-166.

[4] Moor, T., Raisch, J. 2000. Approximation of multiple switched flow systems for the purpose of control synthesis. *Proc. 39th IEEE Conference on Decision and Control*, pp. 3604–3609.

[5] Ramadge, P. J., Wonham, W. M. 1989. The control of discrete event systems. *Proceedings of the IEEE*, vol. 77, pp. 81–98.

[6] Ramadge, P. J., Wonham, W. M. 1989. Modular control of discrete event systems. *Maths. of Control, Signals & Systems*, vol. 1, no. 1, pp. 13–30.

[7] Rudie, K., Wonham, W. M. 1992. Think globally, act locally: decentralized supervisory control. *IEEE Trans. on Automatic Control*, vol. 37, no. 11, pp. 1692–1708.

[8] Willems, J.C. 1989. Models for dynamics, *Dynamics Reported*, vol. 2, pp. 172–269.

[9] Willems, J.C. 1991. Paradigms and puzzles in the theory of dynamic systems, *IEEE Transactions on Automatic Control*, vol. 36, no. 3, pp. 258–294.

[10] Wonham, W.M. 1999. *Notes on control of discrete event systems*. Down-loadable on `http://odin.control.toronto.edu/DES/`