Hybrid control loops, A/D maps, and dynamic specifications

J.M. Davoren[1], T. Moor[1], and A. Nerode[2]

[1] Research School of Information Sciences and Engineering
Australian National University, Canberra ACT 0200 AUSTRALIA
j.m.davoren@anu.edu.au, thomas.moor@anu.edu.au
[2] Department of Mathematics, Cornell University
Ithaca NY 14853 USA
anil@math.cornell.edu

Abstract. We re-examine the basic hybrid control set-up of a continuous plant in a closed feedback loop with a finite state control automaton and an interface consisting of an A/D map and a D/A map. We address the question of how dynamic specifications can be formulated independently of a particular A/D map, and of the effect of refining an A/D map. The main contribution of this paper is that it extends the framework of supervisory controller synthesis for hybrid systems to include more general dynamic specifications, and demonstrates how to employ known results to solve these synthesis problems.

1 Introduction

The basic hybrid control configuration consists of a continuous plant in a closed feedback loop with a finite state supervisory controller, linked by an interface consisting of an A/D map and a D/A map, converting a continuous plant output signal into a discrete controller input signal, and converting a discrete controller output signal into an input to the continuous plant, respectively [16, 12, 7, 5]. Of a particular interest here is the task of controller design; e.g. [7, 16, 9, 4] study classes of control problems in which the continuous plant and D/A map are given, and the task is to construct an A/D map and a supervisory controller so that the closed-loop system fulfills various specifications. Once the A/D map has been constructed, the overall plant exhibits discrete event inputs and outputs. Language inclusion specifications can then be addressed by tools from DES theory (e.g. [14, 15]) and/or within the framework of Willem's behavioural systems theory (e.g. [17]). Consequences for the hybrid control configuration are drawn in [7] and [10], respectively.

In this paper, we re-examine the basic hybrid control configuration and address how control objectives can be stated independently from a particular A/D map, including a discussion on what effects one may expect from refining an A/D map. This matter is of a specific interest whenever a controller synthesis procedure involves A/D map refinement; this is the case in e.g. [7, 16, 4, 9].

On the technical side, we use behavioural systems theory as a framework for our discussion as it cleanly accommodates both motion in continuous space and time, and discrete execution sequences. In that theory, a *dynamical system* is a triple $(T, W, \mathfrak{B})$, where $T \subseteq \mathbb{R}$ is the *time axis*, W is the *signal space*, and $\mathfrak{B} \subseteq W^T := \{f \mid f\colon T \to W\}$ is the *behaviour*. Functions $f\colon T \to W$ are trajectories, and the behaviour $\mathfrak{B}$ is viewed as the set of all trajectories compatible with the phenomena modelled; trajectories not in $\mathfrak{B}$ cannot occur. Typically, a behaviour $\mathfrak{B}$ is defined to be a solution set of a more detailed model; e.g. an ODE for the continuous case. For our purposes, we also consider a behaviour $\mathfrak{B}$ to express *dynamic specifications*, where the trajectories in $\mathfrak{B}$ are those deemed acceptable or permissible for that specification, and those not in $\mathfrak{B}$ are unacceptable or prohibited. With respect to the plant dynamics, our behavioural specifications are similar to the language specifications from DES theory: we ask the closed-loop behaviour to be a subset of the dynamic specification behaviour. However, in contrast to DES theory, we also need to address the continuous aspects of our hybrid control configuration. Therefore, we investigate continuous-time, continuous-space dynamic specifications as behaviours over $T = \mathbb{R}_0^+ := [0, \infty)$ and $X \subseteq \mathbb{R}^n$. The principal case is piecewise-continuous functions $\mathbf{x}\colon \mathbb{R}_0^+ \to X$, since this is what can be generated by a switched plant. For an A/D map from X into Y, with Y finite, we then ask what it means for continuous dynamic specification to be *captured* by a discrete behaviour, over time axis $T = \mathbb{N}$, with words/sequences $\mathbf{y} \in Y^{\mathbb{N}}$, using the A/D map and a language specification. This puts us into a position where we can discuss the effects of A/D map refinement with respect to the task of capturing a given continuous dynamic specification.

The body of the paper is organised as follows. Section 2 consists of mathematical preliminaries. In Section 3, we give a transition system representation of a switched plant coupled with an A/D map, which models the uncontrolled plant when viewed through the lens of the A/D map. In Section 4, we assemble the hybrid closed-loop and show the formal relationship between the hybrid control configuration and hybrid automata models [1, 2]. In Section 5, we formulate the supervisory control problem, and give several illustrative examples of continuous behavioural specifications. Section 6 introduces the notion of a continuous behavioural specification being *captured* by an A/D map together with a discrete behaviour. A/D map refinement as it occurs within supervisory controller synthesis procedures is discussed in Section 7. In Section 8, we show how – in principle – we can solve the control problem for continuous dynamic specifications by using the notion of capturing and drawing from known results on strategic A/D map-refinement and DES-style supervisory controller synthesis.

2 Preliminaries

We adopt the notation from set-valued analysis [3] in writing $r\colon X \rightsquigarrow Y$ to mean $r\colon X \to 2^Y$ is a *set-valued function*, with set-values $r(x) \subseteq Y$ for each $x \in X$, possibly $r(x) = \varnothing$, or equivalently, $r \subseteq X \times Y$ is a *relation*. The *domain* of a

set-valued map is $\mathrm{dom}(r) := \{x \in X \mid r(x) \neq \varnothing\}$. The expressions $y \in r(x)$ and $(x, y) \in r$ are synonymous. Every set-valued map $r\colon X \rightsquigarrow Y$ has an *inverse* or *converse* $r^{-1}\colon Y \rightsquigarrow X$ given by: $x \in r^{-1}(y)$ iff $y \in r(x)$.

A set-valued map $\alpha\colon X \rightsquigarrow Y$ is *total* if $\mathrm{dom}(\alpha) = X$. A total map α defines a *cover* of the set X as follows: for each $y \in Y$, define the set $A_y := \alpha^{-1}(y) \subseteq X$. Then by the totalness condition, we have $X = \bigcup_{y \in Y} A_y$, so the family of sets $\{A_y\}_{y \in Y}$ gives a cover of X. We call the sets A_y the *cells* of the cover α. The cover is *finite* if the *range*, $\mathrm{ran}(\alpha) := \mathrm{dom}(\alpha^{-1})$, is a finite set. A cover α defines an equivalence relation on X of indistinguishability by cover cells: $x \simeq_\alpha x'$ iff $\alpha(x) = \alpha(x')$. A special case of a cover map is when $\alpha\colon X \to Y$ is a (single-valued) total function, in which case the cells A_y for $y \in Y$ are *partition blocks* of the equivalence relation $\simeq_\alpha$.

Given a space $X \subseteq \mathbb{R}^n$, we shall use the term *continuous behaviour* to refer to continuous-time, continuous-space behaviours $\mathfrak{C} \subseteq X^{\mathbb{R}_0^+}$. Note that functions $\mathbf{x}\colon \mathbb{R}_0^+ \to X$ in $\mathfrak{C}$ need not be continuous as maps. Given any set W, we shall use the term *discrete behaviour* to refer to discrete-time behaviours $\mathfrak{B} \subseteq W^{\mathbb{N}}$.

A *transition system* is a structure $\mathcal{S} = (S, W, \delta, S_0)$ where S is a non-empty set of states; W is the external alphabet; $\delta\colon S \times W \rightsquigarrow S$ is the (possibly not deterministic) transition relation; and $S_0 \subseteq S$ is a set of initial states. If $|S| \in \mathbb{N}$ and $|W| \in \mathbb{N}$, then $\mathcal{S}$ is called a *finite state automaton*. Recall that W^* is the set of all finite words over the alphabet W, including the empty word ϵ.

A *state execution sequence* of a transition system $\mathcal{S}$ is a pair of sequences $(\mathbf{s}, \mathbf{w}) \in S^{\mathbb{N}} \times W^{\mathbb{N}}$ or $(\mathbf{s}, \mathbf{w}) \in S^* \times W^*$ such that $\mathbf{s}(0) \in S_0$ and $\mathbf{s}(k+1) \in \delta(\mathbf{s}(k), \mathbf{w}(k))$ for all $k < \mathrm{len}(\mathbf{s})$. A state $s \in S$ is *$\mathcal{S}$-reachable* if there exists a state execution sequence $(\mathbf{s}, \mathbf{w})$ and a $k \leq \mathrm{len}(\mathbf{s})$ such that $s = \mathbf{s}(k)$. A transition system $\mathcal{S}$ has the *non-blocking* property if for every $\mathcal{S}$-reachable state $s \in S$, there exists $w \in W$ such that $\delta(s, w) \neq \varnothing$.

Define the *discrete full state behaviour* of $\mathcal{S}$ to be the set $\mathfrak{B}_{\mathrm{st}}(\mathcal{S}) \subseteq (S^{\mathbb{N}} \times W^{\mathbb{N}})$ of all infinite state execution sequences of $\mathcal{S}$, and the *discrete external behaviour* of $\mathcal{S}$ to be the set $\mathfrak{B}_{\mathrm{ex}}(\mathcal{S}) := \mathcal{P}_W \mathfrak{B}_{\mathrm{st}}(\mathcal{S})$, where $\mathcal{P}_W\colon S \times W \to W$ is the natural projection map. Given a discrete behaviour $\mathfrak{B} \subseteq W^{\mathbb{N}}$, we say that a transition system $\mathcal{S} = (S, W, \delta, S_0)$ is a *state machine realization* of $\mathfrak{B}$, written $\mathcal{S} \cong \mathfrak{B}$, if $\mathfrak{B}_{\mathrm{ex}}(\mathcal{S}) = \mathfrak{B}$. In order to ensure that the restriction to only infinite sequences in the full state behaviour and external behaviour does not result in any loss in the representation of $\mathcal{S}$, care must be taken to ensure that $\mathcal{S}$ is non-blocking.

Let $\mathcal{S} = (S, W, \delta, S_0)$ and $\mathcal{Q} = (Q, W, \gamma, Q_0)$ be two transition systems over a common external alphabet W. Their *synchronous parallel composition* is the system $\mathcal{S} \parallel \mathcal{Q} := (S \times Q, W, \lambda, S_0 \times Q_0)$, where $(s', q') \in \lambda((s, q), w)$ if and only if $s' \in \delta(s, w)$ and $q' \in \gamma(q, w)$.

3 Switched plants and A/D maps

A switched plant is a control system which consists of a finite number of vector fields, with the system switching between one vector field and another. The

control input to a switched plant is via discrete input events which select which vector field is to be active.

Definition 1. *A* switched plant *is a system* $\mathrm{SP} = (U, X, F)$*, where* U *is a finite control (input) alphabet,* $X \subseteq \mathbb{R}^n$ *is the plant state space (equipped with standard Euclidean topology), and* $F\colon U \times X \to \mathbb{R}^n$ *is a function defining a finite family of (time-invariant) differential equations* $\dot{x} = F_u(x)$*, where for each* $u \in U$*, the* u *vector field is* $F_u := F(u, -)\colon X \to \mathbb{R}^n$*.*

For example, a switched plant may arise from a continuous control system $\dot{x} = f(x, v)$ and finitely many state feedback control laws $g_u\colon X \to V$. More generally, one can also consider controllers with their own dynamics, and form a switched plant from finitely many continuous closed-loop systems.

In order to ensure that the state trajectories of a switched plant are well-defined, we assume that the vector fields F_u are locally Lipschitz continuous, and that the state space X is open. Then from each initial condition $x_0 \in X$, each differential equation $\dot{x} = F_u(x)$ has a unique maximal integral curve in X on a well defined maximal interval of time $[0, T_u(x_0))$, where $T_u(x_0) \in \mathbb{R}_0^+ \cup \{\infty\}$. We denote this maximal curve by $\Phi_u(x_0, -)\colon [0, T_u(x_0)) \to X$. In the case of $T_u(x_0) < \infty$, it is well known that $\Phi_u(x_0, -)$ escapes from any *bounded* subset of X at some time less than or equal to $T_u(x_0)$.

Definition 2. *An* A/D map *on a space* X *is a total set-valued map* $\alpha\colon X \rightsquigarrow Y$ *where* Y *is a finite set, with cover cells* $A_y \subseteq X$ *for* $y \in Y$*.*

For any A/D map $\alpha\colon X \rightsquigarrow Y$, we can assume without loss of generality that Y contains a distinguished element $\ddagger \notin \mathrm{ran}(\alpha)$ with the property that $A_\ddagger = \alpha^{-1}(\ddagger) = \varnothing$. In what follows, we use the "dummy" symbol $\ddagger$ as an output symbol indicating that a trajectory will make no more switches.

Definition 3. *Given* $\mathrm{SP} = (U, X, F)$ *and an A/D map* $\alpha\colon X \rightsquigarrow Y$*, we define the transition system model* $\mathcal{S}_{\mathrm{SP}\triangleright\alpha} := (S, W, \delta, S_0)$ *as follows:*

- $S := X \times \mathbb{R}_0^+ \times U \times Y$
- $W := U \times Y$
- *for* $(\mu, \nu) \in U \times Y$*, define:* $(x', \tau', u', y') \in \delta((x, \tau, u, y), (\mu, \nu))$ *iff*
  either ***(i):*** $y \neq \ddagger$ *and* $\nu \neq \ddagger$ *and* $u' = \mu$ *and* $y' = \nu$ *and* $y' \neq y$ *and* $x' = \Phi_\mu(x, \tau' - \tau) \in A_\nu$ *and* $\Phi_\mu(x, t) \in A_y$ *for all* $t \in [0, \tau' - \tau]$*,*
  or ***(ii):*** $y \neq \ddagger$ *and* $\nu = \ddagger$ *and* $u' = \mu$ *and* $y' = \nu$ *and* $\Phi_\mu(x, t) \in A_y$ *for all* $t \in [0, \infty)$*,*
  or else ***(iii):*** $y = \ddagger$ *and* $\nu = \ddagger$ *and* $u' = \mu$ *and* $y' = \nu$*.*
- $S_0 = S$

For an infinite state execution sequence $(\mathbf{s}, \mathbf{w}) \in \mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha})$*, we identify the sequence elements by writing* $\mathbf{s}(i) = (x_i, \tau_i, u_i, y_i)$*, and* $\mathbf{w}(i) = (\mu_i, \nu_i)$*, for each* $i \in \mathbb{N}$*. Let* $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} := \mathfrak{B}_{\mathrm{ex}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha})$*.*

The discrete behaviour $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \subseteq (U \times Y)^{\mathbb{N}}$ is the *uncontrolled* external behaviour of the plant SP with discrete inputs U, when viewed through the lens

of the A/D map α to give discrete outputs Y. The piecewise-continuous state trajectories of the uncontrolled system $(\mathrm{SP} \triangleright \alpha)$ can be recovered from the infinite state execution sequences in $\mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha})$, as follows.

Define a map $\rho\colon \mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha}) \to X^{\mathbb{R}_0^+}$ such that for each $(\mathbf{s}, \mathbf{w}) \in \mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha})$, the function $\rho(\mathbf{s}, \mathbf{w})\colon \mathbb{R}_0^+ \to X$ is given by:

$$\rho(\mathbf{s}, \mathbf{w})(t) \;=\; \Phi_{\mu_i}(x_i, t - \tau_i) \tag{1}$$

for all $i \in \mathbb{N}$, for all $t \in [\tau_i, \tau_{i+1})$ if $\nu_i \neq \ddagger$, and for all $t \in [\tau_i, \infty)$ if $\nu_i = \ddagger$. Then define:

$$\mathfrak{C}_\rho(\mathcal{S}_{\mathrm{SP}\triangleright\alpha}) := \{\rho(\mathbf{s}, \mathbf{w}) \in X^{\mathbb{R}_0^+} \mid (\mathbf{s}, \mathbf{w}) \in \mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha})\} \tag{2}$$

Observe that for a state trajectory $\rho(\mathbf{s}, \mathbf{w})$ of $(\mathrm{SP} \triangleright \alpha)$, the i-th segment during the interval $[\tau_i, \tau_{i+1})$ consists of the flow according to input $\mu_i \in U$ starting from state x_i, with the whole segment lying within the cell A_{y_i}, up to and including the starting point x_{i+1} of the next segment, reached at time τ_{i+1}, and that point x_{i+1} lies in the overlap of cells $A_{y_i} \cap A_{y_{i+1}}$, where $y_{i+1} = \nu_i \in Y$ is the output for stage i.

4 The hybrid closed-loop and hybrid automata models

Given a switched plant $\mathrm{SP} = (U, X, F)$ and an A/D map $\alpha\colon X \rightsquigarrow Y$, the transition system $\mathcal{S}_{\mathrm{SP}\triangleright\alpha}$ over $W = U \times Y$ is able to accept any input events from U without blocking. This property is referred to as *I/S/- plant* form, and technically requires that for all reachable states $s \in S$ and all inputs $u \in U$, there exists an output $y \in Y$ and an $s' \in S$ such that $(s, (u, y), s') \in \delta$. Similarly, a potential controller that is modeled by a transition system $\mathcal{Q} = (Q, U \times Y, \gamma, Q_0)$ is said to be in *I/S/- controller form* if it at any time accepts any output event from Y as generated by the plant. Here the technical requirement is that $\mathcal{Q}$ is non-blocking and that for all reachable states $q \in Q$, for all transitions $(q, (u, y), q') \in \gamma$ and for all controller inputs (plant outputs) $y' \in Y$, there exists $q'' \in Q$ such that $(q, (u, y'), q'') \in \gamma$. Obviously, the parallel composition of a system in I/S/- plant from with one in I/S/- controller form is non-blocking, and this motivates the following definition of admissible supervisory controllers for switched plants:

Definition 4. *Given a switched plant* $\mathrm{SP} = (U, X, F)$ *and A/D map* $\alpha\colon X \rightsquigarrow Y$*, an* admissible supervisory controller *for the uncontrolled system* $(\mathrm{SP} \triangleright \alpha)$ *is a transition system* $\mathcal{Q} = (Q, W, \gamma, Q_0)$ *over* $W = U \times Y$ *that is in I/S/- controller form. The* closed-loop hybrid system *is the transition system* $\mathcal{S}_{\mathrm{SP}\triangleright\alpha} \parallel \mathcal{Q}$*. Let* $\mathfrak{B}_{\mathrm{sup}} := \mathfrak{B}_{\mathrm{ex}}(\mathcal{Q})$*. The discrete external behaviour of the closed-loop system is*

$$\mathfrak{B}_{\mathrm{ex}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha} \parallel \mathcal{Q}) \;=\; \mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \cap \mathfrak{B}_{\mathrm{sup}}$$

From a state execution sequence $(\mathbf{s}, \mathbf{q}, \mathbf{w}) \in \mathfrak{B}_{\mathrm{st}}(\mathcal{S}_{\mathrm{SP}\triangleright\alpha} \parallel \mathcal{Q})$ of the closed-loop, we can recover a piecewise-continuous state trajectory $\rho(\mathbf{s}, \mathbf{q}, \mathbf{w})\colon \mathbb{R}_0^+ \to X$ in the same way as for execution sequences of $\mathcal{S}_{\mathrm{SP}\triangleright\alpha}$. Let $\mathfrak{C}_\rho(\mathcal{S}_{\mathrm{SP}\triangleright\alpha} \parallel \mathcal{Q})$ denote

the set of all piecewise-continuous trajectories recovered from the state execution sequences of the closed-loop hybrid system.

The closed-loop system can be readily shown to be an instance of the standard hybrid automaton model [1, 2].

Definition 5. *A* hybrid automaton *is a system* $\mathcal{H} = (Q, E, X, F, D, R)$ *where:*

- Q *is a finite set of* discrete control modes*;*
- $E\colon Q \rightsquigarrow Q$ *is the* discrete transition relation*;*
- $X \subseteq \mathbb{R}^n$ *is the* continuous state space*;*
- $F\colon Q \times X \to \mathbb{R}^n$ *defines a finite family of* vector fields $F_q : X \to \mathbb{R}^n$*, where* $F_q := F(q, -)$ *for each* $q \in Q$*;*
- $D\colon Q \rightsquigarrow X$ *defines the* mode domain $D_q := D(q) \subseteq X$ *for each* $q \in Q$*;*
- $R\colon X \times E \rightsquigarrow X$ *is the set-valued* reset map.

A function $\mathbf{x}\colon \mathbb{R}_0^+ \to X$ *is a* state trajectory *of* $\mathcal{H}$ *if there exists a discrete index set* $I = \mathbb{N}$ *or* $I = \{0, 1, \ldots, m\}$*, a non-decreasing time-point sequence* $(\tau_i)_{i\in I}$*, with* $\tau_0 = 0$*, a sequence of discrete modes* $(q_i)_{i\in I}$*, and two sequences of continuous states* $(x_i)_{i\in I}$ *and* $(\tilde{x}_i)_{i\in I}$*, the first starting from* $x_0 := \mathbf{x}(0)$*, such that for all* $i \in I$ *and for all* $t \in [\tau_i, \tau_{i+1})$*, the following conditions hold:*

(1.) $\mathbf{x}(t) = \Phi_{q_i}(x_i, t - \tau_i)$ *and* $\mathbf{x}(t) \in D_{q_i}$

(2.) if $i < \sup(I)$ *then* $\tilde{x}_i := \lim_{t\to\tau_{i+1}^-} \Phi_{q_i}(x_i, t - \tau_i)$ *and* $\tilde{x}_i \in D_{q_i}$

(3.) if $i < \sup(I)$ *then* $(q_i, q_{i+1}) \in E$

(4.) if $i < \sup(I)$ *then* $x_{i+1} \in R(\tilde{x}_i, (q_i, q_{i+1}))$

(5.) if $i = \sup(I)$ *then* $\tau_{i+1} = \infty$

Let $\mathfrak{C}(\mathcal{H}) \subseteq X^{\mathbb{R}_0^+}$ *denote the set of all state trajectories of* $\mathcal{H}$*.*

For each discrete transition $(q, q') \in E$, the component reset map is $R_{q,q'} := R(-, q, q') : X \rightsquigarrow X$, and the so-called *guard region* is $G_{q,q'} := \mathrm{dom}(R_{q,q'}) \subseteq X$.

Proposition 1. *Given a closed-loop system formed from* SP*,* α *and* $\mathcal{Q}$*, define the hybrid automaton* $\mathcal{H}(\mathrm{SP}, \alpha, \mathcal{Q}) = (\hat{Q}, E, X, \hat{F}, D, R)$ *as follows:*

- $\hat{Q} := \{(q, u, y) \in Q \times U \times Y \mid (\exists q' \in Q)\,(q, (u, y), q') \in \gamma\,\}$
- $\hat{F}\colon \hat{Q} \times X \to \mathbb{R}^n$ *given by* $\hat{F}((q, u, y), x) = F(u, x)$ *for all* $(q, u, y) \in \hat{Q}$ *and* $x \in X$
- *for each* $(q, u, y) \in \hat{Q}$*, the mode domain* $D_{(q,u,y)} = A_y$
- $E\colon \hat{Q} \rightsquigarrow \hat{Q}$ *given by:*

$$((q, u, y), (q', u', y')) \in E \text{ iff } (q, (u, y), q') \in \gamma \text{ and } A_y \cap A_{y'} \neq \varnothing \tag{3}$$

- *for each* $((q, u, y), (q', u', y')) \in E$*, the reset relation is*

$$R_{(q,u,y),(q',u',y')} := \{(x, x') \in X \times X \mid x \in A_y \cap A_{y'} \text{ and } x' = x\} \tag{4}$$

Then

$$\mathfrak{C}(\mathcal{H}(\mathrm{SP}, \alpha, \mathcal{Q})) = \mathfrak{C}_\rho(\mathcal{S}_{\mathrm{SP}\triangleright\alpha} \parallel \mathcal{Q})$$

This result shows that every hybrid closed-loop can be represented as a hybrid automaton with simple membership-testing resets.

5 Continuous behaviours as dynamic specifications for supervisory controller synthesis

We address the following class of supervisory controller synthesis problems.

Synthesis Problem: *Given a switched plant SP and a continuous behavioural specification $\mathfrak{C}_{\text{spec}}$, construct an A/D map α and a discrete supervisor $\mathcal{Q}$ such that the closed-loop behaviour fulfills the following behavioural inclusion*:

$$\mathfrak{C}_{\rho}(\mathcal{S}_{\text{SP}\triangleright\alpha} \parallel \mathcal{Q}) \subseteq \mathfrak{C}_{\text{spec}}\,. \tag{5}$$

This quite general notion of a continuous dynamic specification gives us a means to place conditions on the evolution of a dynamical system without referring to a model of the system itself; trajectories $x \in \mathfrak{C}_{\text{spec}}$ are deemed acceptable, while trajectories $x \notin \mathfrak{C}_{\text{spec}}$ are deemed unacceptable. To indicate the broad scope of this notion of a specification, we give several illustrative examples.

Example 1: notions of stability. Convergence of trajectories to an equilibrium point $x^\star \in X$ is a necessary condition for asymptotic stability. Consider the continuous behavioual specification:

$$\mathfrak{C}_{\text{conv}} := \{\mathbf{x}\colon \mathbb{R}_0^+ \to X \mid \lim_{t\to\infty} \mathbf{x}(t) = x^\star\} \tag{6}$$

Note that the specification $\mathfrak{C}_{\text{conv}}$ does *not* require $x^\star$ to actually be an equilibrium. This further condition can be expressed by:

$$\mathfrak{C}_{\text{equi}} := \{\mathbf{x}\colon \mathbb{R}_0^+ \to X \mid \mathbf{x}(0) = x^\star \;\Rightarrow\; (\forall t \in \mathbb{R}_0^+)\, \mathbf{x}(t) = x^\star\}\,. \tag{7}$$

Obviously, we can combine the two specifications by taking their intersection: the specification $\mathfrak{C}_{\text{equi}} \cap \mathfrak{C}_{\text{conv}}$ requires $x^\star$ to be an equilibrium to which all trajectories converge.

Example 2: circular motion. An elementary example of hybrid controller synthesis is given in [16], where the control objective is to enforce a clockwise circular motion in the plane $\mathbb{R}^2$. While [16] refers to a particular A/D map in order to formalise this objective, we give an alternative characterisation as a continuous dynamic specification independent of any A/D map. Let $TL := \{l\colon \mathbb{R}_0^+ \to \mathbb{R}_0^+ \mid l \text{ is monotone, unbounded, continuous}\}$ be the set of *time-lag* functions, and consider the clockwise circular reference trajectory $\mathbf{r}\colon \mathbb{R}_0^+ \to \mathbb{R}^2$ given by $\mathbf{r}(t) = (\cos(t), -\sin(t))$. Then define:

$$\mathfrak{C}_{\text{circ}} := \{\, \mathbf{x}\colon \mathbb{R}_0^+ \to X \mid (\exists l \in TL)(\forall t \in \mathbb{R}_0^+)\;\; \mathbf{r}(l(t))^\top \mathbf{x}(t) > 0 \,\} \tag{8}$$

Visually, think of the reference $\mathbf{r}(t)$ as the orthogonal to the separator in a revolving door, rotating clockwise. The "lag" function l allows the revolver to rotate at arbitrary angular velocities, while the inequality ensures that a trajectory $\mathbf{x}$ must stay on the same side of the separator at all times. Intuitively, any person within such a revolving door will be forced to make "steady progress" in a clockwise circular motion, since the separator will only allow "a quarter lap forth and back" relative to the reference trajectory.

Example 3: static safety. The classic form of a safety property consists of specifying a set $Bad \subseteq X$, and requiring that no trajectory ever enters Bad. Consider:

$$\mathfrak{C}_{\text{safe}} := \{\, \mathbf{x}\colon \mathbb{R}_0^+ \to X \mid\ (\forall t \in \mathbb{R}_0^+)\ \mathbf{x}(t) \notin Bad \,\} \tag{9}$$

This type of specification is *static* rather than *dynamic* in the sense that it does not change over time. We will return to these examples after introducing the notion of capturing in the following section.

6 A/D maps and discrete behaviours

Our task here is to formulate the notion of using an A/D map $\alpha\colon X \rightsquigarrow Y$ together with a discrete behaviour $\mathfrak{B} \subseteq Y^{\mathbb{N}}$ to "capture" or "enforce" a continuous dynamic specification $\mathfrak{C}_{\text{spec}} \subseteq X^{\mathbb{R}_0^+}$. Recall that $\ddagger \in Y$ is a distinguished symbol which we use to indicate that no more switches will occur.

Definition 6. *Let* $TP := \{\tau\colon \mathbb{N} \to \mathbb{R}_0^+ \mid \tau(0) = 0 \wedge (\forall i \in \mathbb{N})\, \tau(i) < \tau(i+1)\ \}$ *be the set of* (*strictly increasing*) time-point sequences. *Given an A/D map* $\alpha\colon X \rightsquigarrow Y$ *and a discrete behaviour* $\mathfrak{B} \subseteq Y^{\mathbb{N}}$, *define:*

$$\begin{aligned} \mathcal{C}(\alpha, \mathfrak{B}) \ :=\ \{\, \mathbf{x}\colon \mathbb{R}_0^+ \to X \mid\ & (\exists \mathbf{y} \in \mathfrak{B})(\exists \tau \in TP)(\forall i \in \mathbb{N}) \\ & \big[\ \textit{if}\ \ \mathbf{y}(i) \neq \ddagger\ \ \textit{then}\ \ (\forall t \in [\tau(i), \tau(i+1))\,)\ \mathbf{x}(t) \in A_{\mathbf{y}(i)} \\ & \ \ \textit{and if}\ \ \mathbf{y}(i) \neq \ddagger\ \ \textit{and also}\ \ \mathbf{y}(i+1) = \ddagger \\ & \ \ \textit{then}\ \ (\forall t \in [\tau(i), \infty)\,)\ \mathbf{x}(t) \in A_{\mathbf{y}(i)}\ \big]\ \} \end{aligned} \tag{10}$$

Given a continuous behaviour $\mathfrak{C} \subseteq X^{\mathbb{R}_0^+}$, *we say that the pair* $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}$, *if* $\mathcal{C}(\alpha, \mathfrak{B}) \subseteq \mathfrak{C}$.

The idea is that the continuous behaviour $\mathcal{C}(\alpha, \mathfrak{B})$ includes all and only the trajectories $\mathbf{x}\colon \mathbb{R}_0^+ \to X$ that respect the sequence order of some $\mathbf{y} \in \mathfrak{B}$ considered as a sequence of regions on X via α. To illustrate how an A/D map and a discrete behaviour together capture a continuous behaviour, we continue with our examples.

Ad Example 1: notions of stability. The requirement expressed by $\mathfrak{C}_{\text{conv}}$ depends on the actual topology on X referred to in the expression $\lim_{t\to\infty} \mathbf{x}(t) = x^\star$. Generalising the notion of a limit to cover arbitrary topological spaces (e.g. *Moore-Smith convergence*, [8], §20.IX), the condition is fulfilled if for any open set V containing $x^\star$, there exists a τ such that $\mathbf{x}(t) \in V$ for all $t > \tau$. In the case of the Euclidean topology on X, this cannot be captured by any pair $(\alpha, \mathfrak{B})$ where the signal space Y is finite. However, we can look more broadly at other topologies on X. Fix an A/D map $\alpha\colon X \rightsquigarrow Y$ and consider the finite topology $\mathcal{T}_\alpha \subseteq 2^X$ generated by taking all finite unions and intersections of the α-cells A_y for $y \in Y$. For the most basic case where α defines a finite partition, the open sets in the topology $\mathcal{T}_\alpha$ are just the cells closed under unions. Let $A_{y^\star}$ be the cell such that $x^\star \in A_{y^\star}$, and let:

$$\mathfrak{B}^\alpha_{\text{conv}} := \{\mathbf{y} \in Y^{\mathbb{N}} \mid\ (\exists i \in \mathbb{N})[\ \mathbf{y}(i) = y^\star\ \wedge\ (\forall j > i)\ \mathbf{y}(j) = \ddagger\,\} \tag{11}$$

Then $(\alpha,\, \mathfrak{B}^{\alpha}_{\mathrm{conv}})$ captures $\mathfrak{C}_{\mathrm{conv}}$. For the more general case where α is a finite cover with overlaps, we can also capture $\mathfrak{C}_{\mathrm{conv}}$, but have to replace α with a *refinement* β such that $\mathcal{T}_\beta = \mathcal{T}_\alpha$, where the cells of β are the *join-irreducibles* in $\mathcal{T}_\alpha$ as a lattice of sets (see also [12]). For the equilibrium specification $\mathfrak{C}_{\mathrm{equi}}$, it is also clear that it cannot be captured via any finite range A/D map. However, what can be captured is a weaker version of $\mathfrak{C}_{\mathrm{equi}}$ already relativised to α by replacing true equality $=$ with $\simeq_\alpha$ in Equation (7).

Ad Example 2: circular motion. Consider an A/D map α based on the four quadrants of $\mathbb{R}^2$, similar to [16]. More precisely, let $A_1 = \{(x_1, x_2)|\ x_1 > 0,\ x_2 \geq 0\}$, $A_2 = \{(x_1, x_2)|\ x_1 \leq 0,\ x_2 > 0\}$, $A_3 = \{(x_1, x_2)|\ x_1 < 0,\ x_2 \leq 0\}$, $A_4 = \{(x_1, x_2)|\ x_1 \geq 0,\ x_2 < 0\}$, and, in order to partition the entire $\mathbb{R}^2$, let $A_0 = \{(0, 0)\}$. Denote the corresponding single-valued A/D map by $\alpha\colon Y \to \mathbb{R}^2$, where $Y = \{0, 1, 2, 3, 4\}$. Let

$$\mathfrak{B}^{\alpha}_{\mathrm{circ}} := \ (1432)^\omega \cup (4321)^\omega \cup (3214)^\omega \cup (2143)^\omega \tag{12}$$

Then $(\alpha,\, \mathfrak{B}^{\alpha}_{\mathrm{circ}})$ captures $\mathfrak{C}_{\mathrm{circ}}$.

Ad Example 3: static safety. Consider any A/D map $\alpha\colon X \rightsquigarrow Y$ such that for some $Y_{Bad} \subseteq Y$, we have $Bad \subseteq \bigcup_{y \in Y_{Bad}} A_y$. Then define:

$$\mathfrak{B}^{\alpha}_{\mathrm{safe}} := \ \{\, \mathbf{y}\colon \mathbb{N} \to Y \ \mid\ (\forall i \in \mathbb{N})\ \mathbf{y}(i) \notin Y_{Bad} \,\} \tag{13}$$

Then $(\alpha,\, \mathfrak{B}^{\alpha}_{\mathrm{safe}})$ captures $\mathfrak{C}_{\mathrm{safe}}$.

To resume our study of the notion of *capturing*, fix an A/D-map $\alpha\colon X \rightsquigarrow Y$ and a discrete behaviour $\mathfrak{B} \subseteq Y^{\mathbb{N}}$. It is clear that the set of all dynamic specifications $\mathfrak{C}$ that are captured by the pair $(\alpha, \mathfrak{B})$ forms a complete lattice, with the usual set-theoretic operations. Moreover, $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}_1 \cap \mathfrak{C}_2$ iff $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}_1$ and $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}_2$.

Also observe directly from Definition 6 that the operator $\mathcal{C}(\alpha, \cdot\,)$ distributes over arbitrary unions in the second argument; i.e. if $\mathfrak{B}_i \subseteq Y^{\mathbb{N}}$ for $i \in I$, then: $\mathcal{C}(\alpha, \bigcup_{i\in I} \mathfrak{B}_i) = \bigcup_{i\in I} \mathcal{C}(\alpha, \mathfrak{B}_i)$. Consequently, for a fixed A/D-map and a fixed behaviour $\mathfrak{C} \subseteq X^{\mathbb{R}^+_0}$, the set of all discrete behaviours $\mathfrak{B} \subseteq Y^{\mathbb{N}}$ such that the pair $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}$ forms a complete upper semi-lattice w.r.t. the usual set-theoretic operations. In particular, there uniquely exists a *largest* or *least restrictive* discrete behaviour $\mathfrak{B} \subseteq Y^{\mathbb{N}}$ such that $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}$.

Some immediate consequences of the observed lattice structure are summarized as follows:

Proposition 2. *For any A/D map $\alpha\colon X \rightsquigarrow Y$ and continuous behaviour $\mathfrak{C} \subseteq X^{\mathbb{R}^+_0}$, define:*

$$\mathcal{B}(\alpha, \mathfrak{C}) \ := \bigcup \{\mathfrak{B} \subseteq Y^{\mathbb{N}} \mid \mathcal{C}(\alpha, \mathfrak{B}) \subseteq \mathfrak{C}\,\} \tag{14}$$

Then, for all $\mathfrak{B}' \subseteq Y^{\mathbb{N}}$, we have: $\mathcal{C}(\alpha, \mathfrak{B}') \subseteq \mathfrak{C}$ if and only if $\mathfrak{B}' \subseteq \mathcal{B}(\alpha, \mathfrak{C})$. Furthermore, the following inclusions hold for all $\mathfrak{C}$ and $\mathfrak{B}$:

$$\mathcal{C}(\alpha, \mathcal{B}(\alpha, \mathfrak{C})) \ \subseteq\ \mathfrak{C}\,, \qquad\qquad \mathfrak{B} \subseteq \mathcal{B}(\alpha, \mathcal{C}(\alpha, \mathfrak{B}))\,. \tag{15}$$

7 Refining A/D maps

If the supervisory controller synthesis fails for a given A/D map one may consider a finer A/D-map.

Definition 7. *Let $\alpha\colon X \rightsquigarrow Y$ and $\beta\colon X \rightsquigarrow Z$ be two A/D maps, with cover cells $A_y = \alpha^{-1}(y) \subseteq X$ for $y \in Y$ and $B_z = \beta^{-1}(z) \subseteq X$ for $z \in Z$. We say β is a* refinement *of α, written $\alpha \Subset \beta$, if for each $y \in Y$, there exists $z_1, z_2, \ldots, z_m \in Z$ such that*

$$A_y \;=\; B_{z_1} \cup B_{z_2} \cup \cdots \cup B_{z_m} \tag{16}$$

and for each $z \in Z$, there exists $y \in Y$ such that

$$B_z \;\subseteq\; A_y \tag{17}$$

When $\alpha \Subset \beta$, define a set-valued map $\theta_{\alpha\beta}\colon Y \rightsquigarrow Z$ by: $z \in \theta_{\alpha\beta}(y)$ iff $B_z \subseteq A_y$ or $y = z = \ddagger$. Then $A_y = \bigcup\{B_z \mid z \in \theta_{\alpha\beta}(y)\}$ for all $y \in Y$.

Proposition 3. *Fix a continuous behaviour $\mathfrak{C} \subseteq X^{\mathbb{R}_0^+}$, an A/D map $\alpha\colon X \rightsquigarrow Y$, and a discrete behaviour $\mathfrak{B} \subseteq Y^{\mathbb{N}}$ such that $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}$. Let $DS := \{\ \kappa\colon \mathbb{N} \to \mathbb{N} \mid \kappa(0) = 0 \ \wedge\ (\forall i \in \mathbb{N})\ \kappa(i) < \kappa(i+1)\ \}$ be the set of strictly increasing* discrete-time stretch maps*. Now for any A/D map $\beta\colon X \rightsquigarrow Z$ such that $\alpha \Subset \beta$, define:*

$$\begin{aligned}\mathcal{B}^\beta(\alpha, \mathfrak{B}) := \{\mathbf{z} \in Z^{\mathbb{N}} \mid\ & (\exists \mathbf{y} \in \mathfrak{B})(\exists \kappa \in DS)(\forall j \in \mathbb{N}) \\ & \mathbf{z}|_{[\kappa(j),\kappa(j+1))} \subseteq (\theta_{\alpha\beta}(\mathbf{y}(j)))^*\ \}\,.\end{aligned} \tag{18}$$

Then $(\beta,\ \mathcal{B}^\beta(\alpha, \mathfrak{B}))$ captures $\mathfrak{C}$.

In defining the candidate $\mathcal{B}^\beta(\alpha, \mathfrak{B}) \subseteq Z^{\mathbb{N}}$, we collect all infinite sequences $\mathbf{z}$ that can be decomposed in a sequence of finite words $\mathbf{z}|_{[\kappa(i),\kappa(i+1))}$ such that: (a) each finite word corresponds to a single cover cell $\mathbf{y}(i)$ of α; and that (b) this labelling generates an infinite sequence $\mathbf{y}$ which lies inside the original discrete specification $\mathfrak{B} \subseteq Y^{\mathbb{N}}$.

Proof. To show $(\beta, \mathcal{B}^\beta(\alpha, \mathfrak{B}))$ captures $\mathfrak{C}$, fix an arbitrary $\mathbf{x} \in \mathcal{C}(\beta, \mathcal{B}^\beta(\alpha, \mathfrak{B})))$. Then there exists a sequence $\mathbf{z} \in \mathcal{B}^\beta(\alpha, \mathfrak{B})$ and a $\tau \in TP$ such that for all $i \in \mathbb{N}$, if $\mathbf{z}(i) \neq \ddagger$ then $\mathbf{x}(t) \in B_{\mathbf{z}(i)}$ for all $t \in [\tau(i), \tau(i{+}1))$, and if $\mathbf{z}(i) \neq \ddagger$ but $\mathbf{z}(i{+}1) = \ddagger$ then $\mathbf{x}(t) \in B_{\mathbf{z}(i)}$ for all $t \in [\tau(i), \infty)$. Now by Equation (18), $\mathbf{z} \in \mathcal{B}^\beta(\alpha, \mathfrak{B})$ means there is a witness $\mathbf{y} \in \mathfrak{B}$ and a function $\kappa \in DS$ such that $\mathbf{z}|_{[\kappa(j),\kappa(j+1))} \subseteq (\theta_{\alpha\beta}(\mathbf{y}(j)))^*$ for all $j \in \mathbb{N}$. And from the definition of $\theta_{\alpha\beta}$, we know that $B_{\mathbf{z}(k)} \subseteq A_{\mathbf{y}(j)}$ for all $k \in [\kappa(j), \kappa(j{+}1)) = \{\kappa(j), \kappa(j){+}1, \ldots, \kappa(j{+}1){-}1\}$. We now define a new function $\hat{\tau}\colon \mathbb{N} \to \mathbb{R}_0^+$ by $\hat{\tau}(j) := \tau(\kappa(j))$. Since τ and κ are both strictly increasing, then so is $\hat{\tau}$, and also $\hat{\tau}(0) = \tau(\kappa(0)) = \tau(0) = 0$. Hence $\hat{\tau} \in TP$. We want to show that $\mathbf{x} \in \mathcal{C}(\alpha, \mathfrak{B})$ with witnesses $\mathbf{y} \in \mathfrak{B}$ and $\hat{\tau} \in TP$; then since $(\alpha, \mathfrak{B})$ captures $\mathfrak{C}$, we would have $\mathbf{x} \in \mathfrak{C}$, as required. So now fix any $j \in \mathbb{N}$ and suppose that $\mathbf{y}(j) \neq \ddagger$. Then $\mathbf{z}(k) \neq \ddagger$ for all $k \in [\kappa(j), \kappa(j+1))$. Fix any $t \in [\hat{\tau}(j), \hat{\tau}(j{+}1)) = [\tau(\kappa(j)), \tau(\kappa(j{+}1)))$. Then for some $k \in [\kappa(j), \kappa(j{+}1))$, we

must have $\mathbf{x}(t) \in B_{\mathbf{z}(k)}$, and thus also $\mathbf{x}(t) \in A_{\mathbf{y}(j)}$. For the other case, suppose that $\mathbf{y}(j) \neq \ddagger$ but $\mathbf{y}(j+1) = \ddagger$. Then $\mathbf{z}(k) \neq \ddagger$ and $\mathbf{z}(\kappa(j+1)) = \ddagger$. Fix any $t \in [\hat{\tau}(j), \infty) = [\tau(\kappa(j)), \infty)$. Then we must have $\mathbf{x}(t) \in B_{\mathbf{z}(\kappa(j))}$, and hence $\mathbf{x}(t) \in A_{\mathbf{y}(j)}$, as required. Since $\mathbf{x}$ was arbitrary, we conclude that $(\beta, \mathcal{B}^\beta(\alpha, \mathfrak{B}))$ captures $\mathfrak{C}$.

8 Applying DES and discrete behavioural approaches to controller synthesis

In tackling the general synthesis problem formulated in Section 5, we can build on previous work in [7, 16, 5], first starting by seeking to find an A/D map α and a discrete dynamic specification $\mathfrak{B}^\alpha_{\text{spec}}$ such that $(\alpha, \mathfrak{B}^\alpha_{\text{spec}})$ captures $\mathfrak{C}_{\text{spec}}$. The synthesis problem can then be restated purely in terms of the discrete behaviours, $\mathfrak{B}_{\text{SP}\triangleright\alpha}$ and $\mathfrak{B}^\alpha_{\text{spec}}$: find an admissible discrete supervisor with induced behaviour $\mathfrak{B}_{\text{sup}}$ such that the discrete-time closed-loop behaviour $\mathfrak{B}_{\text{cl}} = \mathfrak{B}_{\text{SP}\triangleright\alpha} \cap \mathfrak{B}_{\text{sup}}$ lies within $\mathfrak{B}^\alpha_{\text{spec}}$. The admissibility requirement for $\mathfrak{B}_{\text{sup}}$ is that it have a transition system realisation in $I/S/-$ controller form. Up to minor notational variations, this restated control problem has been extensively studied in [10, 11, 13]. We provide a terse summary of the main results, in order to show how the broader scope of this contribution relates to the literature.

Fix a switched plant $\text{SP} = (U, X, F)$ and an A/D map $\alpha\colon X \rightsquigarrow Y$, so the induced external behaviour $\mathfrak{B}_{\text{SP}\triangleright\alpha} \subseteq (U \times Y)^{\mathbb{N}}$. Let $\mathfrak{B}^\alpha_{\text{spec}}$ be a discrete dynamic specification. We refer to the pair $(\mathfrak{B}_{\text{SP}\triangleright\alpha},\ \mathfrak{B}^\alpha_{\text{spec}})$ as a *discrete-time supervisory control problem* and ask for an admissible supervisor that enforces $\mathfrak{B}^\alpha_{\text{spec}}$ when interconnected with $\mathfrak{B}_{\text{SP}\triangleright\alpha}$. We give a formal definition of this problem and its solutions.

Definition 8. *Let* $\mathfrak{B}_{\text{sup}} \subseteq (U \times Y)^{\mathbb{N}}$.

- $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}}$ *to said to be* generically implementable *if for all* $k \in \mathbb{N}$, $(\mathbf{u}, \mathbf{y})|_{[0,k]} \in \mathfrak{B}_{\text{sup}}|_{[0,k]}$, $(\tilde{\mathbf{u}}, \tilde{\mathbf{y}})|_{[0,k]} \in W^{k+1}$, $\tilde{\mathbf{u}}|_{[0,k]} = \mathbf{u}|_{[0,k]}$, $\tilde{\mathbf{y}}|_{[0,k)} = \mathbf{y}|_{[0,k)}$ *implies* $(\tilde{\mathbf{u}}, \tilde{\mathbf{y}})|_{[0,k]} \in \mathfrak{B}_{\text{sup}}|_{[0,k]}$.
- *The two behaviours* $\mathfrak{B}_{\text{SP}\triangleright\alpha}$ *and* $\mathfrak{B}_{\text{sup}} \subseteq W^{\mathbb{N}}$ *are said to be* nonconflicting *if* $\mathfrak{B}_{\text{SP}\triangleright\alpha}|_{[0,k]} \cap \mathfrak{B}_{\text{sup}}|_{[0,k]} = (\mathfrak{B}_{\text{SP}\triangleright\alpha} \cap \mathfrak{B}_{\text{sup}})|_{[0,k]}$ *for all* $k \in \mathbb{N}$.
- *The behaviour* $\mathfrak{B}_{\text{sup}}$ *is said to* enforce *the discrete dynamic specification* $\mathfrak{B}^\alpha_{\text{spec}} \subseteq Y^{\mathbb{N}}$ *if* $(\mathbf{u}, \mathbf{y}) \in \mathfrak{B}_{\text{SP}\triangleright\alpha} \cap \mathfrak{B}_{\text{sup}}$ *implies* $\mathbf{y} \in \mathfrak{B}^\alpha_{\text{spec}}$.

The behaviour $\mathfrak{B}_{\text{sup}}$ solves *the control problem* $(\mathfrak{B}_{\text{SP}\triangleright\alpha},\ \mathfrak{B}^\alpha_{\text{spec}})$ *if it satisfies each of these three conditions.* [1]

Note that formally, the trivial behaviour $\mathfrak{B}_{\text{sup}} = \varnothing$ solves $(\mathfrak{B}_{\text{SP}\triangleright\alpha},\ \mathfrak{B}^\alpha_{\text{spec}})$, leading to an empty closed-loop behaviour; obviously, this is undesirable. In response, we ask for the prospective supervisor to be as least restrictive as possible.

[1] The notion here of *generic implementability* corresponds to *implementability w.r.t. a particular plant* as defined in [10], and it can been seen that the alternative formulation leads to precisely the same closed-loop behaviours. The specification $\mathfrak{B}_{\text{spec}}$ in the notation of [10, 11] is related to the $\mathfrak{B}^\alpha_{\text{spec}}$ above by $\mathfrak{B}_{\text{spec}} = \{(\mathbf{u}, \mathbf{y}) |\ \mathbf{y} \in \mathfrak{B}^\alpha_{\text{spec}}\}$

This line of thought is similar to that of DES supervisory control theory [14, 15]. In fact, [10, 11] show that a key result of [14, 15] naturally carries over to the hybrid case: a *least restrictive supervisor* $\mathfrak{B}^{\uparrow}_{\mathrm{sup}}$ that solves $(\mathfrak{B}_{\mathrm{SP}\triangleright\alpha},\ \mathfrak{B}^{\alpha}_{\mathrm{spec}})$ always exists uniquely.

Proposition 4. *The behaviour*

$$\mathfrak{B}^{\uparrow}_{\mathrm{sup}} := \bigcup\{\mathfrak{B}_{\mathrm{sup}} \subseteq (U \times Y)^{\mathbb{N}} |\ \mathfrak{B}_{\mathrm{sup}} \textit{ solves } (\mathfrak{B}_{\mathrm{SP}\triangleright\alpha},\ \mathfrak{B}^{\alpha}_{\mathrm{spec}})\} \tag{19}$$

is a itself a solution of $(\mathfrak{B}_{\mathrm{SP}\triangleright\alpha},\ \mathfrak{B}^{\alpha}_{\mathrm{spec}})$. *We denote the closed-loop behaviour of* $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$ *under least restrictive supervisory control by* $\mathfrak{B}^{\uparrow}_{\mathrm{cl}} := \mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \cap \mathfrak{B}^{\uparrow}_{\mathrm{sup}}$.

In particular, there exists a solution to $(\mathfrak{B}_{\mathrm{SP}\triangleright\alpha},\ \mathfrak{B}^{\alpha}_{\mathrm{spec}})$ with non-empty closed-loop behaviour if and only if $\mathfrak{B}^{\uparrow}_{\mathrm{cl}} \neq \varnothing$.

Unfortunately, very stringent conditions apply to the underlying continuous dynamics Φ when the synthesis is to be carried out directly, based on $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$, or, for that matter, on $\mathcal{S}_{\mathrm{SP}\triangleright\alpha}$ and α. Prospective candidates here are cases in which Φ is linear in both state and time; i.e. straight line evolution, and α a polyhedral partition. On the other hand, if both $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$ and $\mathfrak{B}^{\alpha}_{\mathrm{spec}}$ were realized by finite automata, the least restrictive supervisor could readily be computed drawing from slightly modified methods from DES theory; e.g. [14, 15, 11]. These procedures typically compute a finite automaton realisation of the least restrictive closed-loop $\mathfrak{B}^{\uparrow}_{\mathrm{cl}}$. The latter automaton can also be employed as a supervisor; we give detailed account to this interpretation in [11] from a technical application perspective. A formal conversion to a finite state transition system $\mathcal{Q}$ in I/S/-controller form is straight-forward.

However, while in our framework we may assume $\mathfrak{B}^{\alpha}_{\mathrm{spec}}$ to be realised as a finite automaton, this assumption in general will not hold up for $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$. Consequently, [7, 6, 10, 13] suggest to approximately realize $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$ by a suitable finite automata and then to carry out the synthesis for the problem $(\mathfrak{B}_{\mathrm{ca}},\ \mathfrak{B}^{\alpha}_{\mathrm{spec}})$, where $\mathfrak{B}_{\mathrm{ca}} \subseteq (U \times Y)^{\mathbb{N}}$ denotes the external behaviour induced by the approximate automata realisation. In this approximation-based approach, two main issues present themselves. First, the approximation needs to be sufficiently accurate in order to allow for a successful controller synthesis; we come back to this issue below. Second, assuming that a supervisor could be synthesised for the approximation, one needs to guarantee that desired closed-loop properties are retained when the supervisor is connected to the actual hybrid plant. The second issue is commonly dealt with by requiring that the approximation must be conservative in the sense that it predicts at least all those trajectories on which the actual hybrid plant can evolve. Within our framework this requirement can be stated as the behavioural inclusion $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \subseteq \mathfrak{B}_{\mathrm{ca}}$ and, indeed, this forms a sufficient condition for resolving the second issue:

Proposition 5. *Assume that* $\mathfrak{B}^{\alpha}_{\mathrm{spec}}$, $\mathfrak{B}_{\mathrm{ca}}$ *and* $\mathfrak{B}_{\mathrm{sup}}$ *can all be realised by finite automata. Suppose* $\mathfrak{B}_{\mathrm{sup}}$ *solves the problem* $(\mathfrak{B}_{\mathrm{ca}}, \mathfrak{B}^{\alpha}_{\mathrm{spec}})$ *and suppose* $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \subseteq \mathfrak{B}_{\mathrm{ca}}$. *Then* $\mathfrak{B}_{\mathrm{sup}}$ *is also a solution of* $(\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}, \mathfrak{B}^{\alpha}_{\mathrm{spec}})$. *Furthermore,* $\mathfrak{B}_{\mathrm{ca}} \cap \mathfrak{B}_{\mathrm{sup}} = \varnothing$ *if and only if* $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \cap \mathfrak{B}_{\mathrm{sup}} = \varnothing$.

Recall that on the approximation level, the least restrictive closed-loop behaviour – and hence a realisation $\mathcal{Q}$ in I/S/- controller form – can be computed by methods from DES theory. If $\mathfrak{B}_{\mathrm{sup}} \cong \mathcal{Q}$ is found to enforce a nontrivial closed-loop behaviour $\mathfrak{B}_{\mathrm{ca}} \cap \mathfrak{B}_{\mathrm{sup}} \neq \varnothing$, then by the result above, this realises a solution of our discrete-time supervisory control problem $(\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}, \mathfrak{B}^{\alpha}_{\mathrm{spec}})$. We can conclude from $\mathfrak{B}_{\mathrm{cl}} \subseteq \mathfrak{B}^{\alpha}_{\mathrm{spec}}$ that the hybrid closed loop consisting of SP, α and $\mathcal{Q}$ fulfills the continuous dynamic specification $\mathfrak{C}_{\mathrm{spec}}$ in the sense of Eq. (5), and thus the original control problem given by SP and $\mathfrak{C}_{\mathrm{spec}}$ has also been solved.

If $\mathfrak{B}_{\mathrm{sup}} \cong \mathcal{Q}$ enforces the trivial closed-loop behaviour $\mathfrak{B}_{\mathrm{ca}} \cap \mathfrak{B}_{\mathrm{sup}} = \varnothing$, we distinguish two subcases. First, there may exist no solution for the original problem, in which case we can't complain about failure in finding one. Second, it could be that the chosen A/D map was too coarse and therefore gives a prospective supervisor too little measurement information for it to be able to drive the system according to the continuous dynamic specification. In the latter case, we want to have another attempt with a refined A/D map β, with $\alpha \Subset \beta$. Various methods of A/D map refinement have been discussed in the literature, mostly based on a backward reachability analysis, and the reader is kindly referred to [7, 16, 5, 4]. As worked out in Section 6, discrete dynamic representations of specifications depend on the A/D map. More precisely, when moving from α on to β, we also replace $\mathfrak{B}^{\alpha}_{\mathrm{spec}}$ by $\mathcal{B}^{\beta}(\alpha, \mathfrak{B}^{\alpha}_{\mathrm{spec}})$, as in Eq. 18. In particular, the refinement procedure suggested in [7] for partitions lends itself to the proposed setting.

9 Discussion and conclusion

The framework developed here has two main advantages from the perspective of controller synthesis. First, after a cover refinement, we still refer to the same continuous dynamic specification $\mathfrak{C}_{\mathrm{spec}}$, the latter serving as a formal platform to express the relation between the two control problems stated for α and β, respectively. Thus we make clear that if it is a cover refinement that finally leads to the successful synthesis of a supervisor, then it is in fact the original control objective that we fulfil with our closed-loop design. Second, for any fixed A/D map α, our restated control problem still refers to the "full" hybrid dynamics: although $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha}$ is defined on a discrete-time axis, it refers to the transition system $\mathcal{S}_{\mathrm{SP}\triangleright\alpha}$ over the full hybrid state space, and continuous evolution is recovered by Eq. 1, Section 3. Contrast this with [7], where once the A/D partition is fixed, the treatment of the restated control problem exclusively refers to the so-called *DES Plant*, which is realized as a finite automaton and can be seen to be a rather coarse abstraction. In fact, [13, 10, 11] suggest an ordered family of *l-complete approximations* $\mathfrak{B}_l$, $l \in \mathbb{N}$, where $\mathfrak{B}_{\mathrm{SP}\triangleright\alpha} \subseteq \mathfrak{B}_{l+1} \subseteq \mathfrak{B}_l$, and the DES Plant according to [7] corresponds to the coarsest case $l = 0$. On the other hand, [13, 10, 11] do not discuss the potential gain of accuracy that lies in the refinement of the A/D map. Our current contribution is seen to strategically combine the strengths of both views: the A/D map refinement suggested by [7,

16] as well as the option to increase accuracy for a fixed A/D map suggested by [13, 10, 11].

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.
2. R. Alur, T.A. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, July 2000.
3. J-P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser, Boston, 1990.
4. J. M. Davoren and T. Moor. Logic-based design and synthesis of controllers for hybrid systems. Technical report, RSISE, Australian National University, July 2000. Submitted for publication.
5. J.M. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88:985–1010, July 2000.
6. B. A. Krogh J. E. R. Cury and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transactions on Automatic Control, Special issue on hybrid systems*, 43:564–568, 1998.
7. 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.
8. K. Kuratowski. *Topology*, volume 1. Academic Press, New York, 1966.
9. T. Moor and J. M. Davoren. Robust controller synthesis for hybrid systems using modal logic. In M. D. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 433–446. Springer-Verlag, 2001.
10. T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Systems and Control Letters*, 38:157–166, 1999.
11. T. Moor, J. Raisch, and S. D. O'Young. Discrete supervisory control of hybrid systems based on l-complete approximations. in print, scheduled 2002.
12. A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In R. Grossman *et al.*, editor, *Hybrid Systems*, LNCS 736, pages 297–316. Springer-Verlag, 1993.
13. J. Raisch and S. D. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control, Special issue on hybrid systems*, 43:569–573, 1998.
14. 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.
15. P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.
16. J. A. Stiver, P. J. Antsaklis, and M. D. Lemmon. Interface and controller design for hybrid systems. In P.J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, LNCS 999, pages 462–492. Springer-Verlag, 1995.
17. J. C. Willems. Paradigms and puzzles in the theory of dynamic systems. *IEEE Transactions on Automatic Control*, 36:258–294, 1991.