Discrete Event Dynamic Systems: Theory and Applications, 12, 83–107, 2002

Discrete Supervisory Control of Hybrid Systems Based on *l*-Complete Approximations

THOMAS MOOR thomas.moor@anu.edu.au
Research School of Information Sciences and Engineering, Australian National University, Canberra ACT 0200, Australia

JÖRG RAISCH raisch@mpi-magdeburg.mpg.de
Max-Planck-Institut für Dynamik komplexer technischer Systeme, D-39106 Magdeburg, Federal Republic of Germany

SIU O'YOUNG oyoung@engr.mun.ca
Memorial University of Newfoundland, St. John's, Newfoundland, Canada A1B 3X5

Abstract. The topic of this paper is the synthesis of discrete supervisory control for hybrid systems Σ with discrete external signals. Such systems are in general neither l-complete nor can they be represented by finite state machines. Our solution to the control problem is as follows: we find the strongest l-complete approximation (abstraction) Σ_l for Σ, represent it by a finite state machine, and investigate the control problem for the approximation. If a solution exists on the approximation level, we synthesize the maximally permissive supervisor for Σ_l. We show that it also solves the control problem for the underlying hybrid system Σ. If no solution exists, approximation accuracy can be increased by computing the strongest k-complete abstraction Σ_k, $k > l$. The basic ideas regarding the approximation step are explained within the framework of Willems' behavioral systems theory. Implementation issues are treated in a state space framework, and the main results are interpreted from a traditional control engineering point of view.

Keywords: hybrid systems, supervisory control, behavioral approach, l-complete approximations

1. Introduction

The topic of this paper is supervisory control of time invariant hybrid systems with discrete external (input and output) signals. Roughly speaking, the external behavior (the set of external signals) of such a system is unlikely to possess any properties apart from time invariance. For an example see Lunze (1994), where it is shown that the external behavior of a certain class of particularly simple hybrid systems is Markovian if and only if a very restrictive condition holds. From a more general point of view, we observe that any kind of completeness property that the full (state) behavior possesses will usually be lost when focus is on the external behavior only: in general, the external (discrete) behavior of a hybrid system cannot be represented by a finite state machine. In order to apply supervisory control synthesis techniques, we therefore introduce the strongest l-complete approximation as a discrete abstraction for the hybrid system and represent it by a finite state machine. Similar to the procedure described in Willems (1989), Section 2.4.9, we choose a particularly simple state representation. As an immediate consequence of this

choice of representation, we can explicitly characterize the state evolution law of the approximation in terms of the underlying hybrid system.

Applying a slightly modified version of Ramadge's and Wonham's supervisory control theory (Ramadge and Wonham, 1987; Ramadge and Wonham, 1989), we then check whether the control problem can be solved for the discrete abstraction: can we restrict the (*l*-complete) approximation behavior to a set of ''acceptable'' trajectories? If this is the case, the least restrictive supervisor is determined. It is shown that this supervisor also restricts the external behavior of the hybrid system in the desired fashion.

A good overview of the general area of hybrid systems is given by several conference proceedings volumes (Grossman et al., 1993; Antsaklis et al., 1995; Alur et al., 1996; Antsaklis et al., 1997; Antsaklis et al., 1999) and special journal issues (Antsaklis and Nerode, 1998; Evans and Savkin, 1999).

Control related aspects of hybrid systems have been treated in a number of publications. In the context of the present contribution, approximation based approaches as discussed in Antsaklis et al. (1993); Cury et al. (1998); Chutinan and Krogh (1999); Lunze (1995); Moor (1998a); Raisch and O'Young (1997; 1998b) are most relevant. The present paper is more general than Lunze (1995); Raisch and O'Young (1997; 1998b) as it covers the cases of both ''clock time'' (where the next time instant is determined by the ticking of an external clock) and logic time (where the next time instant is determined by certain discrete events being triggered by the continuous component). It is broader in scope than other logic-time approaches (Moor, 1998a; Chutinan and Krogh, 1999; Tittus and Egardt, 1994), which restrict the continuous part to consist of integrators only.

Like Moor and Raisch (1999b), this paper is based on a previous workshop contribution by the authors (Moor et al., 1998b). Its philosophy is completely different from Moor and Raisch (1999b), however: whereas the latter contribution is set entirely within J.C. Willems' behavioral framework, the present paper employs behavioral systems theory only to provide consistent guidelines for the approximation step. The computational procedure for the approximation step is based on a state representation of the hybrid plant model and generates a state machine as a realization of the approximation. The subsequent controller synthesis step remains entirely within a state space setting. In contrast to the original workshop paper (Moor et al., 1998b), the closed loop scenario is discussed in considerable detail. Additionally, we interpret the main result from a traditional control engineering point of view.

This paper is organized as follows: in Section 2, we give an overview of the supervisory control problem within the framework of Willems' behavioral systems theory. In Section 3, we introduce *l*-complete approximations. In Section 4, we show how to determine the strongest *l*-complete approximation for a given hybrid system, and in Section 5, we apply supervisory control theory to find the maximally permissive feedback controller for this approximation. In Section 6, it is shown that this controller also solves the problem for the underlying hybrid system. Finally, in Section 7, our approximation based approach is applied to a simple but non-trivial example, supervisory temperature control of a three-plate thermal system. Throughout the paper, we also use a trivial (toy) example to illustrate certain key aspects of our approach. To emphasize the fact that our approach covers both clock time and logic (event) time, this latter example is based on an equidistant sampling

grid, while the example in Section 7 uses an asynchronous time axis determined by the occurrence of discrete events.

We stress the point that this paper concentrates on describing a *formal method*: it presents a coherent way to treat control problems for a class of hybrid systems. The method is illustrated by means of two simple examples; more realistic examples have been deliberately omitted. We want to emphasize however, that preliminary versions of our method have been successfully applied to such problems: Klein and Raisch (1998a) deals with the problem of synthesizing a safe emergency-shut-down procedure for a batch evaporator and Klein et al. (1998b) treats the synthesis of an automatic start-up scheme for a distillation column.

2. Supervisory Control in a Behavioral Context

The purpose of this section is to describe the problem of supervisory control within Willems' behavioral systems theory. There, a dynamical system is introduced as a model of a phenomenon with respect to the flow of time.

DEFINITION 1 *See (Willems, 1991), 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 behavior.*

Choosing a time axis T and a signal space W sets up the ''universe'' W^T of all conceivable trajectories for the purpose of modeling. In a ''traditional'' control framework, the external signal space W is usually defined as the product $W = U \times Y$, where U and Y represent the codomains of input and output signal, respectively. Hence, the mechanism of interaction between phenomenon and its environment is the temporal evolution of a trajectory $w \in W^T$. The behavior $\mathfrak{B}$ is then viewed as the set of all trajectories which are compatible with the model: trajectories $w \notin \mathfrak{B}$ cannot occur. Typically, the behavior is the solution set of equations (e.g., differential or difference equations) representing certain aspects of the phenomenon. An overview of this approach is given in Willems (1989; 1991).

In the following, focus is on *discrete behaviors*, i.e., the time axis is $\{t_0, t_1, \ldots\}$ and $|W| \in \mathbb{N}$ (W holds only a finite number of elements) will be assumed throughout this paper. Notice that this set-up covers both the case of ''clock-driven'' and ''event-driven'' systems. In the first case, the progress of time is determined by the ticking of an external clock, and $t_{i+1} - t_i$, $i = 0, 1, \ldots$, is usually constant. In the second case, the time instants t_i are determined by the asynchronous occurrence of discrete events, e.g., by the continuous state variable crossing certain thresholds. To treat both cases within a unified framework, we use an abstract notion of time, representing only the ordering of physical time instances, i.e., $T = \mathbb{N}_0$.

The equations describing $\mathfrak{B}$ may additionally involve latent, or internal, variables, i.e., variables which are not visible from the environment. Therefore, discrete behaviors still cover a typical scenario from the area of hybrid dynamical systems: the model may include continuous dynamics represented by differential equations—hidden from the environment

by quantization. This also includes the case where discrete events are generated by comparing continuous output components with threshold values.

For practical reasons, an efficient representation of $\mathfrak{B}$ is required. In particular, by choosing state machines as means of representation, a link between the behavioral approach and standard terminology from the field of discrete event systems (DES) is provided.

DEFINITION 2 *Let the sets W, $|W| \in \mathbb{N}$, X, $X_0 \subseteq X$, $\delta \subseteq X \times W \times X$ denote the external signal space, the state space, the set of initial conditions and the next state relation, respectively. Then, the tuple $P = (X, W, \delta, X_0)$ is called a state machine and the elements of δ are called transitions. If $|X| \in \mathbb{N}$, P is said to be a finite state machine. The behavior $\mathfrak{B}_s := \{(w,x) \mid (x(t), w(t), x(t+1)) \in \delta \ \forall\ t \in \mathbb{N}_0,\ x(0) \in X_0\}$ is referred to as the induced full behavior, and $\Sigma_s := (\mathbb{N}_0, W \times X, \mathfrak{B}_s)$ as the induced state space system. The external behavior $\mathfrak{B}$ of Σ_s is defined to be the projection of $\mathfrak{B}_s$ onto $W^{\mathbb{N}_0}$, i.e., $\mathfrak{B} := \mathscr{P}_W \mathfrak{B}_s := \{w \mid \exists x : (w,x) \in \mathfrak{B}_s\}$. Vice versa, a state machine P' with induced external behavior $\mathfrak{B}'$ 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 an example, whose *only* purpose is to illustrate certain key aspects of our approach. Hence we choose it to be as simple as possible, although most problems will become trivial for this example. For a nontrivial (but also simple) example, see Section 7. Consider the water tank shown in Figure 1. Its cross sectional area is $F = 100\,\mathrm{cm}^2$, its height $\hat{x} = 30\,\mathrm{cm}$. The attached pump can be switched between two modes: it either feeds water into the tank at a constant rate of $1\,l/\mathrm{min}$, or it removes water from the tank at the same flow rate. The pump is in feed mode if the control input is $u(t) =$ ''+'', and in removal mode if $u(t) =$ ''−''. We work with a fixed sampling rate, l/min, and the control input remains constant between sampling instants. The measurement signal can take two values: $y(t) = E(mpty)$ if the water level $x(t)$ is less or equal to 15 cm, and $y(t) = F(ull)$ if the water level is above 15 cm. Hence $U = \{+, -\}$, $Y = \{E, F\}$, and $X = [0, 30\ \mathrm{cm}]$. The external behavior $\mathfrak{B}$ can be represented by an (infinite) state machine $P = (X, W, \delta, X_0)$, where $W = U \times Y$, $X_0 = X$, $w(t) = (u(t), y(t))$, and $(x(t), w(t), x(t+1)) \in \delta$ if

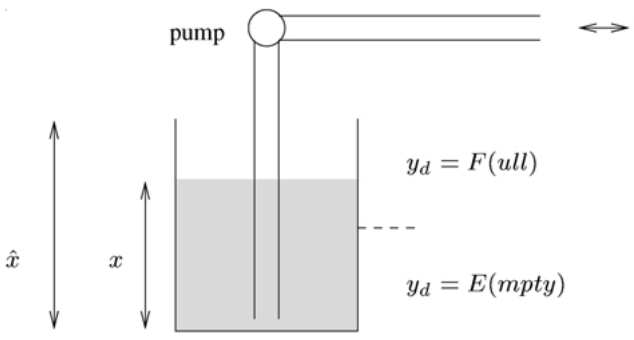

Figure 1. Simple tank example.

$$x(t+1) = f(x(t), u(t)) \tag{1}$$

$$= \begin{cases} x(t) + 10\,\text{cm} & \text{if } u(t) = \text{“}+\text{”} \text{ and } 0 \le x(t) \le 20\,\text{cm}, \\ 30\,\text{cm} & \text{if } u(t) = \text{“}+\text{”} \text{ and } 20\,\text{cm} < x(t) \le 30\,\text{cm}, \\ x(t) - 10\,\text{cm} & \text{if } u(t) = \text{“}-\text{”} \text{ and } 10\,\text{cm} < x(t) \le 30\,\text{cm}, \\ 0\,\text{cm} & \text{if } u(t) = \text{“}-\text{”} \text{ and } 0\,\text{cm} \le x(t) \le 10\,\text{cm}, \end{cases}$$

$$y(t) = g(x(t)) \tag{2}$$

$$= \begin{cases} F & \text{if } 15\,\text{cm} < x(t) \le 30\,\text{cm}, \\ E & \text{if } 0\,\text{cm} \le x(t) \le 15\,\text{cm} \end{cases}$$

Suppose a technical plant is modeled by a dynamical system $\Sigma_p = (\mathbb{N}_0, W, \mathfrak{B}_p)$. To keep notation as simple as possible, we assume that the specifications are also defined on the signal space W. Hence, the set of acceptable trajectories can be characterized by another dynamical system $\Sigma_{spec} = (\mathbb{N}_0, W, \mathfrak{B}_{spec})$. Clearly, if $\mathfrak{B}_p \not\subseteq \mathfrak{B}_{spec}$, the plant behavior needs to be suitably restricted by a supervisory control device: the supervisor's task is to prevent the system Σ from evolving on trajectories which are deemed to be unacceptable. The supervisor will be modeled by a dynamical system $\Sigma_{sup} = (\mathbb{N}_0, W, \mathfrak{B}_{sup})$. Plant and supervisor interact by agreeing on a trajectory $w \in W^T$, hence the closed loop is modeled by $\Sigma_{cl} = (\mathbb{N}_0, W, \mathfrak{B}_{cl})$, where $\mathfrak{B}_{cl} := \mathfrak{B}_p \cap \mathfrak{B}_{sup}$ is the set of all those trajectories which are compatible with both Σ_p and Σ_{sup}. Hence, from the behavioral point of view, supervisory control synthesis is the construction of $\mathfrak{B}_{sup}$ such that $\mathfrak{B}_p \cap \mathfrak{B}_{sup} \subseteq \mathfrak{B}_{spec}$, where $\mathfrak{B}_p$ and $\mathfrak{B}_{spec}$ are given.

So far, the framework is closely related to Ramadge's and Wonham's supervisory control theory (Ramadge and Wonham, 1987; 1989). In fact, the above problem can be fully solved within this theory, if the involved behaviors are realized by *finite* state machines. However, we cannot expect a hybrid dynamical system to be realizable by a finite state machine. Thus, we suggest to construct a conservative approximation $\mathfrak{B}_l$, $\mathfrak{B}_l \supseteq \mathfrak{B}_p$, of the plant behavior $\mathfrak{B}_p$, such that $\mathfrak{B}_l$ can be realized by a finite state machine. Obviously, $\mathfrak{B}_l \cap \mathfrak{B}_{sup} \subseteq \mathfrak{B}_{spec}$ and $\mathfrak{B}_l \supseteq \mathfrak{B}_p$ imply $\mathfrak{B}_p \cap \mathfrak{B}_{sup} \subseteq \mathfrak{B}_{spec}$. Hence, if a supervisor Σ_{sup} restricts Σ_l not to violate Σ_{spec}, then Σ_{sup} will also force Σ_p to obey the specifications. Intuitively, the less accurate the approximation, the more it has to be restricted and the harder the supervisor's task. Vice versa, for a more accurate approximation, we expect the chances for the existence of a suitable supervisor to increase. Figure 2 illustrates the closed loop scenario. A suitable supervisor behavior $\mathfrak{B}_{sup}$ for the original system is indicated by a solid line; a suitable $\mathfrak{B}_{sup}$ for the approximation is indicated by a dashed line.

Although the closed loop behavior $\mathfrak{B}_{cl}$ is well defined by the intersection $\mathfrak{B}_p \cap \mathfrak{B}_{sup}$ certain implementation issues need to be addressed:

1. At any time t, the supervisor may prevent the external signal $w(t)$ from taking certain values. However, in doing so, situations are to be avoided where none of the signal values allowed by the supervisor is compatible with the plant behavior. This is

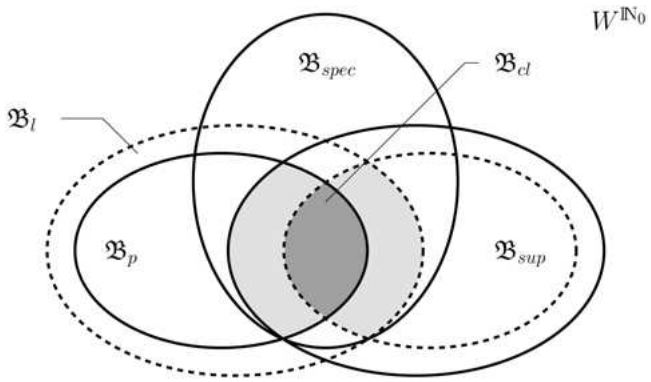

Figure 2. Closed loop scenario.

illustrated by Figure 3, where the past of a closed loop trajectory is denoted by $\bar{w}\,|_{[0,t]}$. Here, $\bar{w}\,|_{[0,t]}$ is the restriction of the map $w : \mathbb{N}_0 \to W$ to the domain $[0, t] \cap \mathbb{N}_0$. The regions $\mathscr{F}_p$ and $\mathscr{F}_{sup}$ indicate the possible future evolution of plant and supervisor, respectively. Then, $\mathscr{F}_p \cap \mathscr{F}_{sup} \neq \emptyset$ is required.

2. In a technical framework, it has to be distinguished between actuators and sensors. Therefore, the signal space W is considered to be the product of a set U of input symbols and a set Y of output symbols, i.e., $W = U \times Y$. Then, the supervisor may only affect the input component in a direct manner, while the output component will be generated by the plant. The product composition of W is motivated by ''traditional''

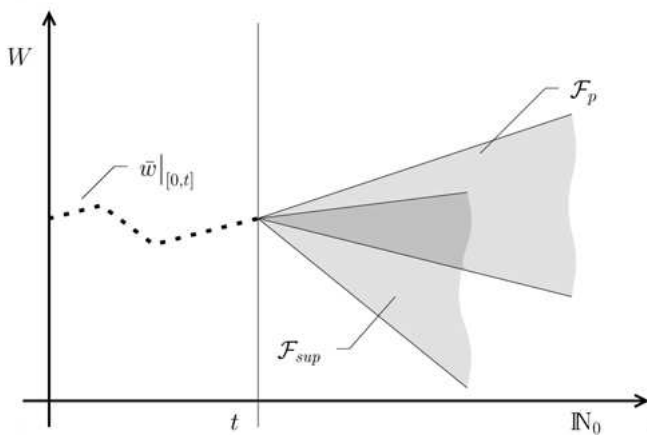

Figure 3. Possible future evolution w. r. t. a known past.

continuous control theory. In Willems (1989; 1991), this is reflected by the notion of I/O systems. Alternatively, as proposed in Ramadge and Wonham (1987; 1989), W can be considered to be the disjoint union of controllable and uncontrollable symbols.

The above is only meant to be an informal discussion. If supervisory control synthesis can be carried out on the approximation level Σ_l, one needs to prove that the desired closed loop properties are retained after connecting the supervisor Σ_{sup} to the underlying hybrid plant model Σ_p. This will be done in Section 6.

3. *l*-Complete Approximations

We propose an approximation scheme that relies on two basic definitions from Willems' ''behavioral approach'': *time invariance* and *l-completeness*. For the reader's convenience, these definitions are collected here. Let σ^t denote the *backwards t-shift*, i.e., $(\sigma^t f)(\tau) := f(t+\tau)$ for all $\tau \in \mathbb{N}_0$, and $\sigma := \sigma^1$. Then:

DEFINITION 3 *See Willems (1991), Def. II.3. A dynamical system* $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ *is said to be time invariant if* $\sigma\mathfrak{B} \subseteq \mathfrak{B}$.

Implicitly, a system is uniquely determined by its behavior; we therefore refer to a *behavior* as being time invariant, if it belongs to a time invariant *system*. This convention is also used with respect to all properties defined in the sequel.

A sufficient (but not necessary) condition for time invariance of the external behavior induced by a state machine $P = (X, W, \delta, X_0)$ is $X = X_0$; i.e., the initial conditions are not restricted.

DEFINITION 4 *See Willems (1991), Def. II.4. Let* $l \in \mathbb{N}$. *A time invariant dynamical system* $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ *is said to be l-complete if*

$$w \in \mathfrak{B} \iff \sigma^t w|_{[0,l]} \in \mathfrak{B}|_{[0,l]} \forall\ t \in \mathbb{N}_0 \tag{3}$$

To keep notation reasonably compact, we do not distinguish between $w\,|_{[t_1,t_2]} \in W^{[t_1,t_2]}$ and the string of symbols $\langle w(t_1), \ldots w(t_2)\rangle \in W^{t_2-t_1+1}$. Note that shifting is defined to be of higher priority than restricting: $\sigma^t w\,|_{[0,l]} = (\sigma^t w)\,|_{[0,l]} = w\,|_{[t,t+l]}$.

For any time invariant system, the left hand side of (3) implies the right hand side. Hence, the crucial point about l-completeness is the converse direction. If $\mathfrak{B}$ is known to be l-complete, Property (3) allows, in principle, to test whether a trajectory $w \in W^{\mathbb{N}_0}$ is an element of $\mathfrak{B}$: a window with length $l+1$ is moved along the time axis, selecting finite strings $w|_{[t,t+l]}$, $t \in \mathbb{N}_0$. If all these strings are contained in $\mathfrak{B}|_{[0,l]}$, then $w \in \mathfrak{B}$. This scenario is illustrated by Figure 4. Note that $\mathfrak{B}|_{[0,l]}$ is a finite set as W is assumed to be finite.

Implementing this procedure by a state machine leads to a realization of an l-complete system. Here, the state variable z is defined to memorize the last l values of the external signal: $z(t) := \langle w(t-l), \ldots\ w(t-1)\rangle \in Z_l$, $t \geq l$. A transition to a next state $z(t+1)$ can

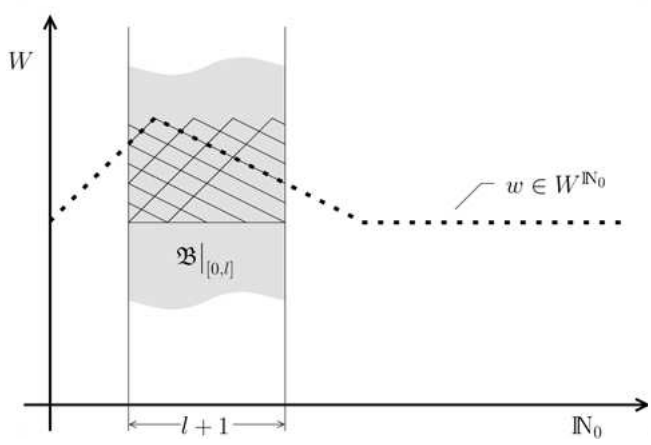

Figure 4. Testing for $w \in B$.

take place if and only if this does not contradict $\mathfrak{B}|_{[0,l]}$; i.e., $\langle w(t-l), \ldots\ w(t)\rangle \in \mathfrak{B}|_{[0,l]}$. This construction is similar to Willems (1989), Section 2.4.9. However, since our time axis is $\mathbb{N}_0$, we need to take into account the effect of shorter strings for $t < l$. This is formalized by the following theorem.

THEOREM 1 *Let* $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ *be an l-complete time invariant system. Let*

$$Z_l := \{\omega^*\} \cup_{1 \le r \le l} W^r, \quad Z_0 = \{\omega^*\} \tag{4}$$

where $\omega^* \notin W$ *is a new ''dummy'' symbol meaning ''no external signal present so far''. Let*

$$\delta_l := \cup_{0 \le r \le l}\ \delta_l^r \subseteq Z_l \times W \times Z_l \tag{5}$$

where

$$\delta_l^0 := \{(\omega^*, \omega_0, \omega_0) |\ \langle \omega_0 \rangle \in \mathfrak{B}|_{[0,0]}\}, \tag{6}$$

$$\delta_l^r := \{(\langle \omega_0, \ldots\ \omega_{r-1}\rangle, \omega_r, \langle \omega_0, \ldots\ \omega_r\rangle) |\ \langle \omega_0, \ldots\ \omega_r\rangle \in \mathfrak{B}|_{[0,r]}\}\ 1 \le r < l, \tag{7}$$

$$\delta_l^l := \{(\langle \omega_0, \ldots\ \omega_{l-1}\rangle, \omega_l, \langle \omega_1, \ldots\ \omega_l\rangle) |\ \langle \omega_0, \ldots\ \omega_l\rangle \in \mathfrak{B}|_{[0,l]}\} \tag{8}$$

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

Proof: Let $\mathfrak{B}_s$ denote the full behavior induced by P_l. We need to show $\mathfrak{B} = \mathfrak{B}_{ex} := \mathscr{P}_W \mathfrak{B}_s$. Choose an arbitrary but fixed $w \in W^{\mathbb{N}_0}$ and let

$$z(t) := \begin{cases} \omega^* & \text{if } t = 0, \\ \langle w(0), \ldots, w(t-1) \rangle & \text{if } 0 < t < l, \\ \langle w(t-l), \ldots, w(t-1) \rangle & \text{if } t \geq l \end{cases} \tag{9}$$

In order to prove $w \in \mathfrak{B}_{ex} \Leftrightarrow w \in \mathfrak{B}$ we first assume $w \in \mathfrak{B}_{ex}$. Hence there must exist a z' such that $(w, z') \in \mathfrak{B}_s$. From the definition of δ_l it follows by induction that $z = z'$ and therefore $(z(t), w(t), z(t+1)) \in \delta_l$ for all $t \in \mathbb{N}_0$. Furthermore, the definition of δ_l implies $\langle z(t), w(t) \rangle \in \mathfrak{B}|_{[0,l]}$ for all $t \geq l$. Observe by Equation (9), $w|_{[t-l,t]} = \langle z(t), w(t) \rangle \in \mathfrak{B}|_{[0,l]}$ for all $t \geq l$. Since $\mathfrak{B}$ is l-complete, this implies $w \in \mathfrak{B}$. We now assume $w \in \mathfrak{B}$. It is obvious that $(z(t), w(t), z(t+1)) \in \delta_l$ for all $t \in \mathbb{N}_0$, and $z(0) = \omega^* \in Z_0$. Hence $(w, z) \in \mathfrak{B}_s$ and therefore $w \in \mathfrak{B}_{ex}$.

As $|W|$ is assumed to be finite throughout this paper, the state space of P_l is also finite. Not all systems are l-complete, however. For a system $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ without this property, we now propose the notion of a ''strongest l-complete approximation''. Roughly speaking, this is a system evolving on the same time axis $\mathbb{N}_0$ and within the same signal space W as the original system, and with the smallest l-complete behavior that covers the ''original'' behavior $\mathfrak{B}$. Formally, this can be written as:

DEFINITION 5 *Let $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ and $\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l)$ be time invariant dynamical systems, with $l \in \mathbb{N}$. Σ_l is said to be a strongest l-complete approximation induced by Σ if the following conditions hold:*

i. $\mathfrak{B}_l \supseteq \mathfrak{B}, \mathfrak{B}_l$ *is l-complete*

ii. $\mathfrak{B}'_l \supseteq \mathfrak{B}, \mathfrak{B}'_l$ *is l-complete* $\Longrightarrow \mathfrak{B}'_l \supseteq \mathfrak{B}_l$

The motivation for Definition 5 is the following: we want to synthesize supervisory control for Σ on the basis of the approximation Σ_l. Clearly, we need condition (i) to hold; otherwise, $\mathfrak{B}$ could contain unacceptable trajectories which could not be predicted by the approximation Σ_l and hence not be suppressed by a control strategy based on the approximate model. It is also obvious that we want condition (ii) to hold: the smaller $\mathfrak{B}_l$, the more accurate the approximation Σ_l, and the better the chances for a suitable supervisor to exist. See also Figure 2.

PROPOSITION 1 *Let $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ be a time invariant dynamical system. Choose an arbitrary $l \in \mathbb{N}$. Then, the strongest l-complete approximation induced by Σ, denoted by $\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l)$, exists uniquely, and $\mathfrak{B}_l$ is given by:*

$$\mathfrak{B}_l = \{w |\ w \in W^T,\ \sigma^t w|_{[0,l]} \in \mathfrak{B}|_{[0,l]}\ \forall\ t \in \mathbb{N}_0\} \tag{10}$$

Proof: Uniqueness follows immediately from the definition. To prove existence, take $\mathfrak{B}_l$ as defined by (10) and check conditions (i) and (ii). Σ is time invariant, hence $w \in \mathfrak{B} \Rightarrow \sigma^t w|_{[0,l]} \in \mathfrak{B}|_{[0,l]}$ for all $t \in \mathbb{N}_0$, and therefore $\mathfrak{B}_l \supseteq \mathfrak{B}$. l-completeness of $\mathfrak{B}_l$ is obvious, hence (i) holds. Now, take any l-complete $\mathfrak{B}'_l$ that satisfies $\mathfrak{B}'_l \supseteq \mathfrak{B}$. Pick any

$w \in \mathfrak{B}_l$; from (10), it follows immediately that $\sigma^t w|_{[0,l]} \in \mathfrak{B}|_{[0,l]} \subseteq \mathfrak{B}'_l|_{[0,l]}$ for all $t \in \mathbb{N}_0$. $\mathfrak{B}'_l$ being l-complete implies $w \in \mathfrak{B}'_l$. Hence, $\mathfrak{B}'_l \supseteq \mathfrak{B}_l$, and existence has been proven. ■

Corollary 1 is an immediate consequence of equation (10):

COROLLARY 1 *Let* $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ *be a time invariant dynamical system and* $\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l)$ *the induced strongest l-complete approximation. Then,*

i. $\mathfrak{B}_l|_{[0,l]} = \mathfrak{B}|_{[0,l]}$,

ii. $\mathfrak{B}_l \supseteq \mathfrak{B}_{l+1} \supseteq \mathfrak{B}$,

iii. $\Sigma_l = \Sigma \Longleftrightarrow \Sigma$ *is l-complete.*

By (i) and Theorem 1, a finite state machine realizing the approximation Σ_l can be constructed in terms of the restriction $\mathfrak{B}|_{[0,l]}$ of the original behavior $\mathfrak{B}$. Note that the former is a finite set. Therefore, l-complete approximations can be established for any dynamical system which exhibits a time invariant discrete external behavior $\mathfrak{B}$ such that $\mathfrak{B}|_{[0,l]}$ can be computed.

By (ii), the strongest k-complete approximation $\mathfrak{B}_k$, $k > l$, is guaranteed to be at least as accurate as the strongest l-complete approximation $\mathfrak{B}_l$.

By (iii), we know that the strongest l-complete approximation $\mathfrak{B}_l$ is exact if and only if the original behavior $\mathfrak{B}$ is also l-complete.

4. Hybrid State Space Systems

We now apply the results from above to a class of time invariant hybrid systems. It is characterized by the fact that the external behavior is discrete (i.e., $T = \mathbb{N}_0, |W| \in \mathbb{N}$), while the state set X forms a subset of $\mathbb{R}^n \times D$, $|D| \in \mathbb{N}$. Thus, we still restrict systems to be discrete time. However, from our point of view, it does not matter whether the time axis $\mathbb{N}_0$ is ''clock time'' (e.g., a regular sampling grid) or ''logic time'', enumerating the occurrence of events (where events could be defined as certain continuous variables crossing certain threshold values). As suggested in Section 2, the external signal space W is considered to be the product $W = U \times Y$, where U, $|U| \in \mathbb{N}$, is a set of input symbols and Y, $|Y| \in \mathbb{N}$, is a set of output symbols. By writing $(u, y) = w \in W^{\mathbb{N}_0}$, we refer to this product composition, i.e., we implicitly assume $u = \mathscr{P}_U w \in U^{\mathbb{N}_0}$, $y = \mathscr{P}_Y w \in Y^{\mathbb{N}_0}$, where $\mathscr{P}_U$ and $\mathscr{P}_Y$ denote the canonical projections from $(U \times Y)^{\mathbb{N}_0}$ onto $U^{\mathbb{N}_0}$ and $Y^{\mathbb{N}_0}$, respectively (or from $U \times Y$ onto U and Y). The product composition $W = U \times Y$ shall be reflected by the structure of the next state relation, as defined below:

DEFINITION 6. *A state machine* $P = (X, W, \delta, X_0)$ *is said to be a hybrid state machine, if* $X \subseteq \mathbb{R}^n \times D$, $|D| \in \mathbb{N}$. *A state machine* $P = (X, W, \delta, X_0)$, $W = U \times Y$, *is said to be an I/S/O machine if for every* $\xi \in X$ *and* $\nu \in U$ *there uniquely exist* $\mu \in Y$ *and* $\xi' \in X$ *such that* $(\xi, (\nu, \mu), \xi') \in \delta$.

To guarantee time invariance of the induced state space system, the initial conditions are assumed to be not restricted throughout this section, i.e., $X_0 = X$. Then, the external behavior $\mathfrak{B} = \mathscr{P}_W \mathfrak{B}_s$ induced by P is also time invariant, hence the strongest l-complete approximation $\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l)$ for $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$ exists. Note that we cannot expect the external behavior $\mathfrak{B}$ to possess any completeness property. Therefore, $\mathfrak{B}_l = \mathfrak{B}$ will in general not be possible, and some degree of model accuracy will be lost when approximating $\mathfrak{B}$ by $\mathfrak{B}_l$. On the other hand, we know from the previous section that $\mathfrak{B}_l$ can be realized by the finite state machine P_l and is hence amenable to standard methods from the field of DES theory.

We now discuss how to compute $\mathfrak{B}|_{[0,l]} = \mathfrak{B}_l|_{[0,l]}$ for a given hybrid state machine P. As a first step, sets of compatible states are defined; this provides a link between finite strings $w|_{[0,l]}$ of the external signal w and the internal state evolution.

DEFINITION 7. *For a given state machine $P = (X, W, \delta, X)$ with induced state behavior $\mathfrak{B}_s$, let $\mathscr{X}(\bar{w}|_{[0,l]}) \subseteq X$ denote the set of all states at time $t = l$ that are compatible with $\bar{w}|_{[0,l]} \in W^{l+1}$:*

$$\mathscr{X}(\bar{w}|_{[0,l]}) := \{\xi |\ \exists\, (w, x) \in \mathfrak{B}_s :\ x(l) = \xi,\ w|_{[0,l]} = \bar{w}|_{[0,l]}\} \tag{11}$$

Clearly, a given string $\bar{w}|_{[0,l]} \in W^{l+1}$ is an element of $\mathfrak{B}|_{[0,l]}$ if and only if there exist states compatible with $\bar{w}|_{[0,l]}$, i.e., if $\mathscr{X}(\bar{w}|_{[0,l]}) \neq \emptyset$. On the other hand, if the state space system induced by P is trim (see Willems, 1991, page 270), the sets of compatible states can be derived by the iterative formula given in the following proposition. Loosely speaking, trimness requires that every element of the state set is ''hit'' by a trajectory of the full state behavior, i.e., $\forall \xi \in X \exists\, (w, x) \in \mathfrak{B}_s,\ t \in \mathbb{N}_0 :\ x(t) = \xi$.

PROPOSITION 2 *Let $P = (X, W, \delta, X)$ be a state machine with a trim induced state space system. Then, for any trajectory $\bar{w} \in W^{\mathbb{N}_0}$ the following holds:*

$$\mathscr{X}(\bar{w}|_{[0,0]}) = \{\xi |\ \exists\, \xi^+ \in X :\ (\xi, \bar{w}(0), \xi^+) \in \delta\} \tag{12}$$

$$\mathscr{X}(\bar{w}|_{[0,l+1]}) = \{\xi |\ \exists\, \xi^- \in \mathscr{X}(\bar{w}|_{[0,l]}) :\ (\xi^-, \bar{w}(l), \xi) \in \delta\} \cap \mathscr{X}(\bar{w}(l+1)) \tag{13}$$

Proof: It is obvious that any ξ in one of the left hand side sets in (12) and (13) satisfies the conditions stated on the respective right hand side. Hence the left hand side sets are contained in the right hand side sets. To show the converse, pick any ξ from the right hand side set of equation (12). Trimness and time invariance imply that there exists a trajectory $(w^+, x^+) \in \mathfrak{B}_s, x^+(0) = \xi^+$. Define $(w, x) \in (W \times X)^{\mathbb{N}_0}$ by $x(0) := \xi, w(0) := \bar{w}(0)$, and $x(t) := x^+(t-1)$, $w(t) := w^+(t-1)$ for all $t \geq 1$. Then, $(x(t), w(t), x(t+1)) \in \delta$ for all $t \in \mathbb{N}_0$. Hence $(w, x) \in \mathfrak{B}_s$, and therefore $\xi = x(0) \in \mathscr{X}(\bar{w}|_{[0,0]})$, yielding equation (12). Now, pick any ξ from the right hand side set in equation (13). As $\xi^- \in \mathscr{X}(\bar{w}|_{[0,l]})$, we know a trajectory $(w^-, x^-) \in \mathfrak{B}_s$ to exist such that $w^-|_{[0,l]} = \bar{w}|_{[0,l]}$ and $x^-(l) = \xi^-$. As $\xi \in \mathscr{X}(\bar{w}(l+1))$, there exists a trajectory $(w^+, x^+) \in \mathfrak{B}_s$ such that $x^+(0) = \xi$ and $w^+(0) = \bar{w}(l+1)$. In analogy to the previous case, we construct a trajectory $(w, x) \in \mathfrak{B}_s$

by concatenating $(w^-, x^-)|_{[0,l]}$ and (w^+, x^+): define $(w, x) \in (W \times X)^{\mathbb{N}_0}$ by $x(t) := x^-(t)$, $w(t) := w^-(t-1)$ for all $t \leq l$, and $x(t) := x^+(t-l-1)$, $w(t) := w^+(t-l-1)$ for all $t > l$. This yields $\xi \in \mathscr{X}(\bar{w}|_{[0,l+1]})$. Hence, it has been shown that equation (13) holds. ■

In the case of I/S/O machines, Definition 6 implies the existence of maps $f : X \times U \to X$ and $g : X \times U \to Y$ such that

$$\delta = \{(\xi, (\nu, \mu), \xi') |\ \xi' = f(\xi, \nu),\ \mu = g(\xi, \nu)\} \tag{14}$$

Thus, I/S/O machines with unrestricted initial conditions $X_0 = X$ are trim. In this situation, Proposition 2 yields for any $u \in U^{\mathbb{N}_0}$, $y \in Y^{\mathbb{N}_0}$:

$$\mathscr{X}((u, y)\big|_{[0,0]}) = g^{-1}_{u(0)}(y(0)) \tag{15}$$

$$\mathscr{X}((u, y)\big|_{[0,l+1]}) = f(\mathscr{X}((u, y)\big|_{[0,l]}), u(l)) \cap g^{-1}_{u(l+1)}(y(l+1)) \tag{16}$$

Here, g_ν^{-1}, $\nu \in U$, denotes the inverse image of $g(\nu, \cdot)$, i.e., $g_\nu^{-1}(\mu) := \{\xi |\ g(\xi, \nu) = \mu\}$. Whenever one is able to repeatedly compute images under f, inverse images under g, and intersections of those, the above equations can be used to compute the sets of compatible states and hence the finite set $\mathfrak{B}|_{[0,l]}$. The next state relation δ_l is then set up according to equations (5)–(8), leading to the finite state machine $P_l = (Z_l, W, \delta_l, Z_0)$.

A number of similar approximation schemes have been proposed in the literature, namely ''abstraction A_l'' (Raisch, 1998a; Raisch and O'Young, 1997), ''condensed model of order r'' (Moor, 1998a), ''approximating automaton $\mathscr{A}_N$'' (Cury et al., 1998). In contrast to these references, we have made extensive use of J.C. Willems' behavioral systems theory. This point of view enabled the rather general discussion given in Section 3: recall that so far, apart from time invariance, we have only imposed two assumptions on the structure of the plant, namely that its external behavior is discrete and that there are control inputs and measured outputs. Both assumptions are natural when investigating the problem of synthesizing a purely discrete supervisor. Our results are neither restricted to ''clock time'', as are Raisch (1998a); Raisch and O'Young (1997), nor to ''logic time'' as Cury et al. (1998); Moor (1998a); we allow the underlying plant model to contain both continuous and discrete components (Cury et al., 1998; Moor, 1998a assume it to be purely continuous); finally, our discussion is not confined to the case where the continuous dynamics consists of integrators only—we also consider the (computationally most challenging) combination of a non-trivial continuous flow and ''logic time''. In the latter case, f and g are given in implicit form only, and estimates of the sets of compatible states have to be established by an additional approximation procedure as discussed in Moor and Raisch (1998c). Here, the crucial point is to conservatively estimate the sets of states reachable at the time the next external event occurs. This problem is not specific to supervisory control synthesis, but is fundamental for the analysis of hybrid systems in general. Extensive discussions for various classes of non-trivial continuous flows can be

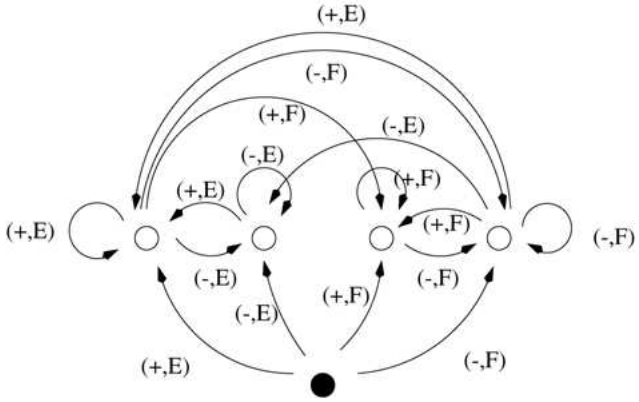

Figure 5. Realization P_1 for approximation Σ_1.

found in the literature; see Chutinan and Krogh (1998), Dang and Maler (1998), Puri et al. (1996).

Procedures for the computation of P_l are provided in detail by: Raisch and O'Young (1997) for sampled linear systems with quantized external variables, Moor (1998a) for switched integrator systems and Moor and Raisch (1999a) for switched linear systems.

Before using the strongest l-complete approximation Σ_l for the purposes of supervisory control synthesis, we summarize the proposed abstraction procedure and apply it to the simple tank example introduced in Section 2: our starting point is a hybrid state machine $P = (X, W, \delta, X)$ with induced full behavior $\mathfrak{B}_s$ and (discrete) external behavior $\mathfrak{B} = \mathscr{P}_W \mathfrak{B}_s$. First, choose an $l \in \mathbb{N}$ and compute the sets of compatible states $\mathscr{X}(\bar{w}|_{[0,r]})$ for all strings $\bar{w}|_{[0,r]}$, $r \leq l$. This can be done by a recursive formula as stated in Proposition

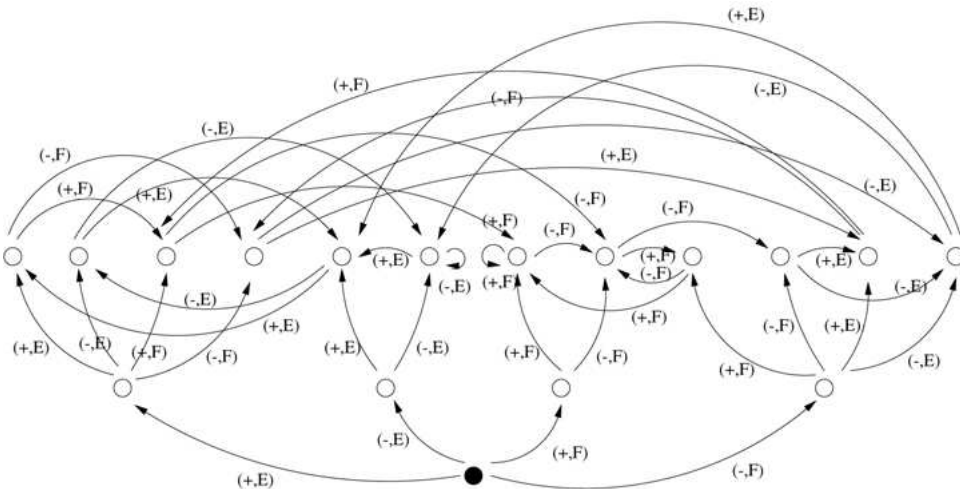

Figure 6. Realization P_2 for approximation Σ_2.

2. Then, $\mathfrak{B}|_{[0,l]}$ is the set of strings $\bar{w}|_{[0,l]}$ with $\mathscr{X}(\bar{w}|_{[0,l]}) \neq \emptyset$. The (purely discrete) next state relation δ_l is set up according to equations (5)–(8). From Theorem 1, we know that $P_l = (Z_l, W, \delta_l, Z_0)$ is a realization of the strongest l-complete approximation Σ_l induced by $\Sigma = (\mathbb{N}_0, W, \mathfrak{B})$, hence $\mathfrak{B}_l \supseteq \mathfrak{B}$. Recall that the latter is a necessary condition if controller synthesis for Σ is to be based on Σ_l.

We now illustrate the proposed abstraction procedure by applying it to the tank example from Section 2: to build the strongest 1-complete approximation Σ_1, we need to check for all strings $\bar{w}\big|_{[0,0]}, \bar{w}\big|_{[0,1]}$ whether $\mathscr{X}(\bar{w}\big|_{[0,0]})$ and $\mathscr{X}(\bar{w}\big|_{[0,1]})$ are non-empty. From (15) and the continuous model equations (1), (2), it follows immediately that

$$\mathscr{X}((+,E)) = \mathscr{X}((-,E)) = g^{-1}(E) = [0,15]$$
$$\mathscr{X}((+,F)) = \mathscr{X}((-,F)) = g^{-1}(F) = (15,30]$$

Note that in our example the output map g does not depend on the input symbol. The units for the interval boundaries are cm. Using equation (16), we get

$$\mathscr{X}((+,E),(+,E)) = \mathscr{X}((+,E),(-,E)) = [10,25] \cap [0,15] = [10,15],$$
$$\mathscr{X}((+,E),(+,F)) = \mathscr{X}((+,E),(-,F)) = [10,25] \cap (15,30] = (15,25],$$
$$\mathscr{X}((-,E),(+,E)) = \mathscr{X}((-,E),(-,E)) = [0,5] \cap [0,15] = [0,5],$$
$$\mathscr{X}((-,E),(+,F)) = \mathscr{X}((-,E),(-,F)) = [0,5] \cap (15,30] = \emptyset,$$
$$\mathscr{X}((+,F),(+,F)) = \mathscr{X}((+,F),(-,F)) = (25,30] \cap (15,30] = (25,30],$$
$$\mathscr{X}((+,F),(+,E)) = \mathscr{X}((+,F),(-,E)) = (25,30] \cap [0,15] = \emptyset,$$
$$\mathscr{X}((-,F),(+,F)) = \mathscr{X}((-,F),(-,F)) = (5,20] \cap (15,30] = (15,20],$$
$$\mathscr{X}((-,F),(+,E)) = \mathscr{X}((-,F),(-,E)) = (5,20] \cap [0,15] = (5,15]$$

Then, using Theorem 1, we can easily set up a realization P_1 for Σ_1. It is shown in Figure 5, where the initial state is represented by a filled circle.

To build a finite state machine P_2 representing the strongest 2-complete approximation Σ_2, we also need to check non-emptiness of $\mathscr{X}(\bar{w}\big|_{[0,2]})$ for all strings $\bar{w}\big|_{[0,2]}$. We omit this rather boring exercise and show the resulting P_2 in Figure 6. By construction, $\mathfrak{B}_2 \subseteq \mathfrak{B}_1$. In our case, the inclusion is strict, i.e., there exist trajectories $w \in \mathfrak{B}_1$ that are not contained in $\mathfrak{B}_2$. As an example, take $w = (+,E)\,(+,E)\,(+,E)\,\ldots$. Hence approximation Σ_1 deems is possible that water can be added to the tank for an abritrary period of time without ever measuring $F(ull)$. On the basis of approximation Σ_2, we can tell that this cannot happen. In fact, the reader can easily check that $\Sigma_2 = \Sigma$, implying that our toy example is 2-complete.

5. Supervisory Control of State Machines

In Section 2, the problem of supervisory control has been motivated within a behavioral framework. In the following, we present a solution on the realization level. Hence, focus is on approximations and specifications which are realized by finite state machines:

$$\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l) \cong P_l = (Z_l, W, \delta_l, Z_0) \tag{17}$$

$$\Sigma_{spec} = (\mathbb{N}_0, W, \mathfrak{B}_{spec}) \cong P_{spec} = (X_{spec}, W, \delta_{spec}, X_{spec_0}) \tag{18}$$

We now introduce some basic terminology related to state machines:

DEFINITION 8 *Let $P_a = (A, W, \alpha, A_0)$ and $P_b = (B, W, \beta, B_0)$ be state machines.*

- *Reachability: 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 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.*

- *Don't care symbol ''$-$'': We use ''$(a, w, -) \in \alpha$'' as an abbreviation for ''$(a, w, a') \in \alpha$ for some $a' \in A$''; in analogy, ''$(a, w, -) \notin \alpha$'' means ''$(a, w, a') \notin \alpha$ for all $a' \in A$''.*

- *Nonblocking property: The state machine P_a is called temporally nonblocking, if for every reachable state $a \in A$, there exists $w \in W$ such that $(a, w, -) \in \alpha$.*

- *Subset: P_a is said to be a subset of P_b if $A \subseteq B$, $\alpha \subseteq \beta$ and $A_0 \subseteq B_0$. This is denoted by $P_a \subseteq P_b$.*

- *Union: $P_a \cup P_b := (A \cup B, W, \alpha \cup \beta, A_0 \cup B_0)$.*

- *Parallel composition: $P_a \parallel P_b := (A \times B, W, \lambda, A_0 \times B_0)$, where $((a, b), w, (a', b')) \in \lambda$ if and only if $(a, w, a') \in \alpha$ and $(b, w, b') \in \beta$.*

Our solution procedure for the supervisory control synthesis problem is as follows: we first synthesize a supervisor for an l-complete approximation $\Sigma_l = (\mathbb{N}_0, W, \mathfrak{B}_l)$ by employing a modified version of Ramadge's and Wonham's theory (Ramadge and Wonham, 1987; 1989); a version similar to the one presented here has been described in Raisch and O'Young (1998b). Then, we show that the supervisor obtained for Σ_l does indeed solve the problem for the underlying hybrid system.

Formally, all unacceptable trajectories are removed by intersecting $\mathfrak{B}_l$ and $\mathfrak{B}_{spec}$. It is a well known fact that the intersection of two behaviors can be realized by the parallel composition of the realizations of the two behaviors, i.e., $(\mathbb{N}_0, W, \mathfrak{B}_l \cap \mathfrak{B}_{spec}) \cong P_{\parallel} := P_l \parallel P_{spec} = (Q, W, \lambda, Q_0)$. Thus, forming the parallel composition removes all transitions which violate the specifications—but this is done without caring for implementability: Items 1 and 2 from Section 2 need to be considered. Item 1 requires that the closed loop realization is nonblocking. Item 2 requires that a transition (ζ, ω, ζ') can only be eliminated if all partner transitions $(\zeta, \omega', \zeta'')$, $\mathscr{P}_U \omega' = \mathscr{P}_U \omega$, are eliminated simultaneously. The optimal supervisor's job can then be thought of as enforcing the ''least restrictive'' but implementable subset of $P_l \parallel P_{spec}$. This is formalized in the following paragraph:

DEFINITION 9 *Two transitions* $\delta^1 = (\zeta_1, \omega_1, \zeta_1') \in \delta_l$ *and* $\delta^2 = (\zeta_2, \omega_2, \zeta_2') \in \delta_l$ *of* P_l *are called partners, if* $\zeta_1 = \zeta_2$ *and* $\mathscr{P}_U\omega_1 = \mathscr{P}_U\omega_2$. *A state machine*

$$\tilde{P} = (\tilde{Q}, W, \tilde{\lambda}, \tilde{Q}_0) \subseteq P_l \parallel P_{spec} = (Q, W, \lambda, Q_0) \tag{19}$$

is called a controllable substructure of $P_l \parallel P_{spec}$ *w. r. t.* P_l, *if for every reachable state* (ζ, ξ_{spec}) *of* $\tilde{P}$, *a transition* $((\zeta, \xi_{spec}), \omega, (\zeta', \xi'_{spec})) \in \lambda$ *can only be an element in* $\tilde{\lambda}$, *if* $((\zeta, \xi_{spec}), \omega', (\zeta'', -)) \in \tilde{\lambda}$ *for every partner* $(\zeta, \omega', \zeta'')$ *of* (ζ, ω, ζ').

The relation $\subseteq$ induces a partial ordering on the set of all controllable substructures of $P_l \parallel P_{spec}$. As nonreachable states do not affect the induced behavior, the following investigations can be restricted to reachable state machines.

LEMMA 1 *Let* $\{\tilde{P}_{CN}\}$ *be the set of all controllable substructures of* $P_l \parallel P_{spec}$ *w. r. t.* P_l *which are both reachable and temporally nonblocking. Then* $\{\tilde{P}_{CN}\}$ *is closed under union.*

Proof: Let $\tilde{P}_1 = (\tilde{Q}_1, W, \tilde{\lambda}_1, \tilde{Q}_{1_0}), \tilde{P}_2 = (\tilde{Q}_2, W, \tilde{\lambda}_2, \tilde{Q}_{2_0}) \in \{\tilde{P}_{CN}\}$. Obviously, $\tilde{P}_1 \cup \tilde{P}_2$ is reachable. Let ρ be any state of $\tilde{P}_1 \cup \tilde{P}_2$. Then either $\rho \in \tilde{Q}_1$ is a (reachable) state of $\tilde{P}_1$, hence there exist $\omega \in W$ and $\rho' \in \tilde{Q}_1$ such that $(\rho, \omega, \rho') \in \tilde{\lambda}_1$; or $\rho \in \tilde{Q}_2$ is a (reachable) state of $\tilde{P}_2$, hence there exist $\omega \in W$ and $\rho' \in \tilde{Q}_2$ such that $(\rho, \omega, \rho') \in \tilde{\lambda}_2$. Existence of a transition $(\rho, w, \rho') \in \tilde{\lambda}_1 \cup \tilde{\lambda}_2$ is therefore guaranteed, and $\tilde{P}_1 \cup \tilde{P}_2$ is temporally nonblocking. Now, focus on a state (ζ, ξ_{spec}) of $\tilde{P}_1 \cup \tilde{P}_2$ and a transition $\gamma := ((\zeta, \xi_{spec}), \omega, (\zeta', \xi'_{spec})) \in \tilde{\lambda}_1 \cup \tilde{\lambda}_2$. Then, either $\gamma \in \tilde{\lambda}_1$ or $\gamma \in \tilde{\lambda}_2$. As both $\tilde{P}_1$ and $\tilde{P}_2$ are reachable controllable substructures of $P_l \parallel P_{spec}$ w. r. t. P_l, for every partner $(\zeta, \omega', \zeta'')$ of (ζ, ω, ζ') there exists a transition $((\zeta, \xi_{spec}), \omega', (\zeta'', -))$ either in $\tilde{\lambda}_1$ or $\tilde{\lambda}_2$. Thus, $\tilde{P}_1 \cup \tilde{P}_2$ is a controllable substructure of $P_l \parallel P_{spec}$ w. r. t. P_l. ■

Hence, if non-empty, $\{\tilde{P}_{CN}\}$ forms an upper-semilattice (with the join operation being $\cup$). Clearly, $\{\tilde{P}_{CN}\}$ is finite. Therefore, the following holds:

COROLLARY 2 *If* $\{\tilde{P}_{CN}\}$ *is non-empty, there exists a (unique) greatest subset of* $P_l \parallel P_{spec}$ *(w. r. t. the ordering via* $\subseteq$*) which is*

i. a controllable substructure of $P_l \parallel P_{spec}$ *w. r. t.* P_l,

ii. reachable,

iii. temporally nonblocking.

If $\{\tilde{P}_{CN}\}$ is empty, the supervisory control problem has no solution. This implies that either the strongest l-complete approximation Σ_l is too coarse, or the specifications are too strict (they cannot be met no matter how accurate our approximation is) and need to be relaxed. In the former case, we need to provide a finer approximation: we can try the

strongest k-complete approximation $\Sigma_k,\ k > l$, which, by Corollary 1, is guaranteed to be at least as accurate as Σ_l.

If $\{\tilde{P}_{CN}\}$ is non-empty, denote its supremal element by $P^+_{sup} = (Q^+_{sup}, W, \lambda^+_{sup}, Q^+_{sup_0})$. It can be interpreted as the transition structure of $P_l \parallel P_{spec}$ that survives under the least restrictive implementable supervisory control policy which guarantees the specifications to be met. It can also be interpreted as a realization of the least restrictive supervisor, which, at every state $\rho \in Q^+_{sup}$, disables certain symbols from $W = U \times Y$. P^+_{sup} can be formally synthesized via a fixed-point algorithm in a computer-aided design environment. This procedure has been coded in $C++$ with an object oriented architecture (O'Young, 1998).

As P^+_{sup} is least restrictive, we do not expect transitions to be uniquely determined: the input symbols permitted in state $\rho \in Q^+_{sup}$ form a set $\mathscr{U}(\rho) := \{\nu |\ (\rho, (\nu, -), -) \in \lambda^+_{sup}\} \subseteq U$. As P^+_{sup} is reachable and temporally nonblocking, $\mathscr{U}(\rho)$ holds at least one element for every state $\rho \in Q^+_{sup}$. However, within a technical framework, actuators apply a unique control symbol to the plant. Therefore, a selection mechanism needs to be implemented. Formally, this is represented by a map $\mathscr{S}: 2^U \to U$, where $\mathscr{S}(\mathscr{U}) \in \mathscr{U}$ for all $\mathscr{U} \subseteq U$ and 2^U denotes the set of all subsets of U. When the supervisor is in state $\rho \in Q^+_{sup}$, the input symbol $\nu := \mathscr{S}(\mathscr{U}(\rho))$ will be applied. The combination of supervisor P^+_{sup} and selection mechanism $\mathscr{S}$ can be implemented as a finite state machine $P_{sup} = (Q_{sup}, W, \lambda_{sup}, Q_{sup_0}) \subseteq P^+_{sup}$: let

$$\tilde{\lambda} := \{(\rho, \omega, \rho') |\ (\rho, \omega, \rho') \in \lambda^+_{sup},\ \mathscr{P}_U \omega = \mathscr{S}(\mathscr{U}(\rho))\} \tag{20}$$

Then, state space, next state relation and initial states are given by

$$Q_{sup} := \{\rho |\ \rho \text{ is a reachable state of} (Q^+_{sup}, W, \tilde{\lambda}, Q^+_{sup_0})\} \tag{21}$$

$$\lambda_{sup} := \{(\rho, \omega, \rho') |\ (\rho, \omega, \rho') \in \tilde{\lambda},\ \rho \in Q_{sup}\} \tag{22}$$

$$Q_{sup_0} := Q^+_{sup_0} \tag{23}$$

Clearly, P_{sup} is reachable and temporally nonblocking. Furthermore, observe P_{sup} to be a controllable substructure of $P_l \parallel P_{spec}$ w. r. t. P_l. Hence, P_{sup} is an element of $\{\tilde{P}_{CN}\}$, and therefore, when connected with P_l, satisfies our implementability criteria. Denote the induced dynamical system by $\Sigma_{sup} = (\mathbb{N}_0, W, \mathfrak{B}_{sup})$, $\Sigma_{sup} \cong P_{sup}$. By construction, Σ_{sup} does not exhibit any unacceptable trajectories:

$$\emptyset \subset \mathfrak{B}_{sup} \subseteq \mathfrak{B}_l \cap \mathfrak{B}_{spec} \tag{24}$$

We now turn again to the simple tank example described in Sections 2 and 4. Recall that in this example the progress of time is determined by the ''ticking'' of an external clock. Suppose the closed loop specifications are as follows: after two time steps, the output symbol $E(mpty)$ is not allowed to occur any more, i.e., $y(t) \stackrel{!}{=} F(ull)$ for $k \geq 3$. A state machine realizing the specification behavior is shown in Figure 7.

It is obvious that the given specification cannot be enforced on the basis of

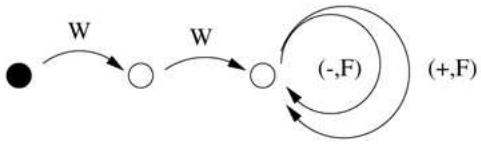

Figure 7. Realization P_{spec} for specification Σ_{spec}.

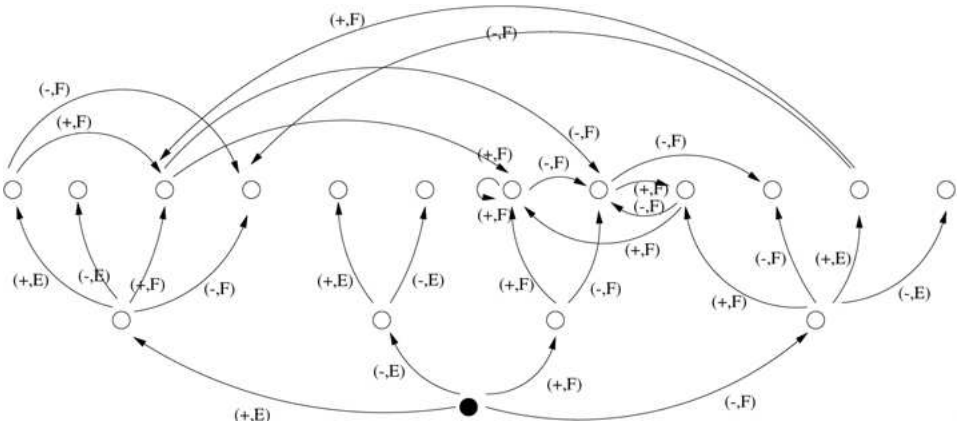

Figure 8. Reachable part of $P_2 P_{spec}$.

approximation Σ_1: recall that $w = (+,E)(+,E)(+,E) \ldots \in \mathfrak{B}_1$; hence, as far as Σ_1 is concerned, even keeping the input constantly at ''+'' cannot rule out the *E(mpty)*-symbol after any specified time. Formally, this is reflected in the fact that $\{\tilde{P}_{CN}\}$ is empty for $P_1 \parallel P_{spec}$.

We now turn to the approximation Σ_2. Figure 8 shows the reachable part of $P_2 \parallel P_{spec}$. In the next step, we determine P^+_{sup}, i.e., the supremal element in $\{\tilde{P}_{CN}\}$ (the set of all controllable substructures of $P_2 \parallel P_{spec}$ w. r. t. P_2 which are both reachable and temporally nonblocking). P^+_{sup}, the least restrictive supervisor, is shown in Figure 9. It tracks the

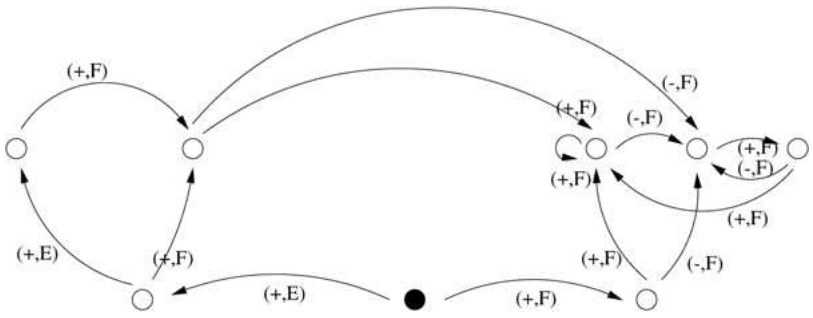

Figure 9. Least restrictive supervisor P^+_{sup}.

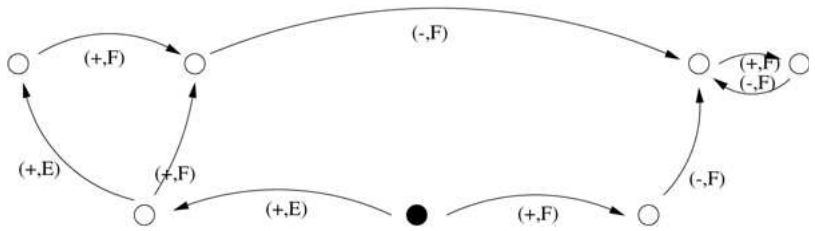

Figure 10. P_{sup}, consisting of P^{+}_{sup} and selection meachanism $\mathscr{S}$.

external signal, and implements a straightforward control strategy: if the symbol observed last is $w(t) = (+, F)$, i.e., if a measurement $y(t) = F(ull)$ is followed by $u(t) =$''+'', the next control symbol $u(t+1)$ can either be ''+'' or ''−''. Otherwise, only ''+'' is allowed. To complete the example, we specify the selection meachanism $\mathscr{S}$ by $\mathscr{S}(\{-, +\}) =$''−''. The resulting supervisor P_{sup} (Figure 10) will then simply choose input symbol ''−'' whenever P^{+}_{sup} leaves a choice between ''+'' and ''−''.

6. Applying Supervisory Control to the Plant

Throughout this section we assume $\{\tilde{P}_{CN}\}$ to be non-empty, i.e., the l-complete approximation Σ_l is accurate enough and a suitable supervisor exists. We then need to answer the following question: what is going to happen when we ''connect'' the supervisory controller $\Sigma_{sup} \cong P_{sup}$ to the underlying (hybrid) plant model $\Sigma_p \cong P$? As discussed in Section 2, the closed loop behavior $\mathfrak{B}_{cl}$ is the intersection of $\mathfrak{B}_p$ and $\mathfrak{B}_{sup}$. Hence, $\mathfrak{B}_{sup} \subseteq \mathfrak{B}_{spec}$ (equation (24)) and the plant Σ_p under supervision Σ_{sup} will not exhibit any unacceptable behavior; i.e., $\mathfrak{B}_{cl} = \mathfrak{B}_p \cap \mathfrak{B}_{sup} \subseteq \mathfrak{B}_{spec}$, see also Figure 2. In Section 5, we have shown that P_{sup}, when connected to the approximation P_l, satisfies the implementability criteria from Section 5. This result carries over to $P_{cl} = P \parallel P_{sup}$: we will prove that P_{cl} is a temporally nonblocking controllable substructure of $P \parallel P_{spec}$ w. r. t. P.

In addition to the notation (17) and (18), we use the following nomenclature:

$$\Sigma_p = (\mathbb{N}_0, W, \mathfrak{B}_p) \cong P = (X, W, \delta, X) \tag{25}$$

$$\Sigma_{sup} = (\mathbb{N}_0, W, \mathfrak{B}_{sup}) \cong P_{sup} = (Q_{sup}, W, \lambda_{sup}, Q_{sup_0}) \tag{26}$$

$$\Sigma_{cl} = (\mathbb{N}_0, W, \mathfrak{B}_{cl}) \cong P_{cl} = (Q_{cl}, W, \lambda_{cl}, Q_{cl_0}) \tag{27}$$

As in Section 4, the state space system $\Sigma_s = (\mathbb{N}_0, W \times X, \mathfrak{B}_s)$ induced by P is assumed to be trim. Furthermore, to guarantee time invariance, the initial conditions $X_0 = X$ are not restricted. Additionally, P is required to be an I/S/O machine.

Note that the closed loop $P_{cl} := P \parallel P_{sup}$ is a subset of $P \parallel P_l \parallel P_{spec}$. In a first step, we therefore investigate $P \parallel P_l$. As $\mathfrak{B} \subseteq \mathfrak{B}_l$, we expect P not to be restricted by the interconnection with its approximation P_l. This is indeed the case:

PROPOSITION 3 *Let* $(\xi, \zeta) \in X \times Z_l$ *be a reachable state of* $P \parallel P_l$ *and* $\omega \in W$ *such that* $(\xi, \omega, -) \in \delta$. *Then*$(\zeta, \omega, -) \in \delta_l$.

Proof: As (ξ, ζ) is reachable, there must exist $t \in \mathbb{N}_0$, $\bar{w} \in W^{\mathbb{N}_0}$, $\bar{x} = X^{\mathbb{N}_0}$, $\bar{z} = Z_l^{\mathbb{N}_0}$ such that $\bar{z}(0) \in Z_0 = \{\omega^\star\}$, $\bar{z}(t) = \zeta$, $\bar{x}(t) = \xi$ and $(\bar{x}(\tau), \bar{w}(\tau), \bar{x}(\tau + 1)) \in \delta$, $(\bar{z}(\tau), \bar{w}(\tau), \bar{z}(\tau + 1)) \in \delta_l$ for all $\tau < t$. Trimness and time invariance of Σ_s imply the existence of $(w, x) \in \mathfrak{B}_s$ such that $w(t) = \omega$, $w|_{[0,t-1]} = \bar{w}|_{[0,t-1]}$, $x|_{[0,t]} = \bar{x}|_{[0,t]}$. From the definition of δ_l in Theorem 1 we conclude that either $\zeta = \omega^\star$ if $t = 0$, or $\zeta = w|_{[0,t-1]}$ if $0 < t < l$, or $\zeta = w|_{[t-l,t-1]}$ if $t \geq l$, respectively. In the last case, let $\zeta' := w|_{[t-l+1,t]}$, otherwise let $\zeta' := w|_{[0,t]}$. Then $(\zeta, \omega, \zeta') \in \delta_l$. ■

As an immediate consequence, $P \parallel P_l$ is temporally nonblocking. With this preliminary result, we can now check our implementability criteria:

COROLLARY 3 *The closed loop realization* $P_{cl} = P \parallel P_{sup}$ *is temporally nonblocking.*

Proof: Let (ξ, ζ, ξ_{spec}) be a reachable state of $P \parallel P_{sup}$. Observe (ξ, ζ) and (ζ, ξ_{spec}) to be reachable states of $P \parallel P_l$ and P_{sup}, respectively. As P_{sup} is temporally nonblocking, there exist $\omega \in W$ and $\zeta' \in Z_l$ such that $((\zeta, \xi_{spec}), \omega, (\zeta', -)) \in \lambda_{sup}$. As P is an I/S/O machine, we can pick an $\omega' \in W, \mathscr{P}_U \omega' = \mathscr{P}_U \omega$, such that $(\xi, \omega', \xi'') \in \delta$. Hence, by Proposition 3, $(\zeta, \omega', \zeta'') \in \delta_l$ for some $\zeta'' \in Z_l$. Note that $(\zeta, \omega', \zeta'')$ and (ζ, ω, ζ') are partners. Thus, P_{sup} being a controllable substructure of $P_l \parallel P_{spec}$ w. r. t. P_l implies $((\zeta, \xi_{spec}), \omega', (\zeta'', -)) \in \lambda_{sup}$. Therefore, $((\xi, \zeta, \xi_{spec}), \omega', (\xi'', \zeta'', -)) \in \lambda_{cl}$. ■

COROLLARY 4 *The closed loop realization* $P_{cl} = P \parallel P_{sup}$ *is a controllable substructure w. r. t. P.*

Proof: Let (ξ, ζ, ξ_{spec}) be a reachable state of $P \parallel P_{sup}$. Let $\omega \in W$ and $(\xi', \zeta', \xi'_{spec}) \in X \times Z_l \times X_{spec}$ such that $((\xi, \zeta, \xi_{spec}), \omega, (\xi', \zeta', \xi'_{spec})) \in \lambda_{cl}$. Let $(\xi, \omega', \xi'') \in \delta$ be a partner of $(\xi, \omega, \xi') \in \delta$. We then need to show that $((\xi, \zeta, \xi_{spec}), \omega', (\xi'', -, -)) \in \lambda_{cl}$. Observe that $((\zeta, \xi_{spec}), \omega, (\zeta', \xi'_{spec})) \in \lambda_{sup}$, and, by Proposition 3, $(\zeta, \omega, \zeta') \in \delta_l, (\zeta, \omega', \zeta'') \in \delta_l$. Recall, that P_{sup} is a controllable substructure of $P_l \parallel P_{spec}$ w. r. t. P_l. Hence, $((\zeta, \xi_{spec}), \omega', (\zeta'', -)) \in \lambda_{sup}$. This implies $((\xi, \zeta, \xi_{spec}), \omega', (\xi'', \zeta'', -)) \in \lambda_{cl}$. ■

In order to emphasize the relevance of our implementability criteria, we now investigate the closed loop scenario from a traditional control engineering point of view. This is illustrated by Figure 11.

At time $t \in \mathbb{N}_0$, let the closed loop realization $P_{cl} = P \parallel P_{sup}$ be in the (reachable) state $(\xi, \rho) \in X \times Q_{sup}$; hence, the plant P and the supervisor P_{sup} are in the reachable states ξ and ρ, respectively. Recall that P_{sup} is temporally nonblocking. Hence, there exists a transition $(\rho, -, -) \in \lambda_{sup}$. The input symbol $\nu := \mathscr{S}(\mathscr{U}(\rho))$ will be applied to the plant P. As the plant is assumed to be an I/S/O machine, there exists a unique output symbol $\nu \in Y$ and an unique next state $\xi' \in X$ such that $(\xi, (\nu, \mu), \xi') \in \delta$. Thus, at time t the value of

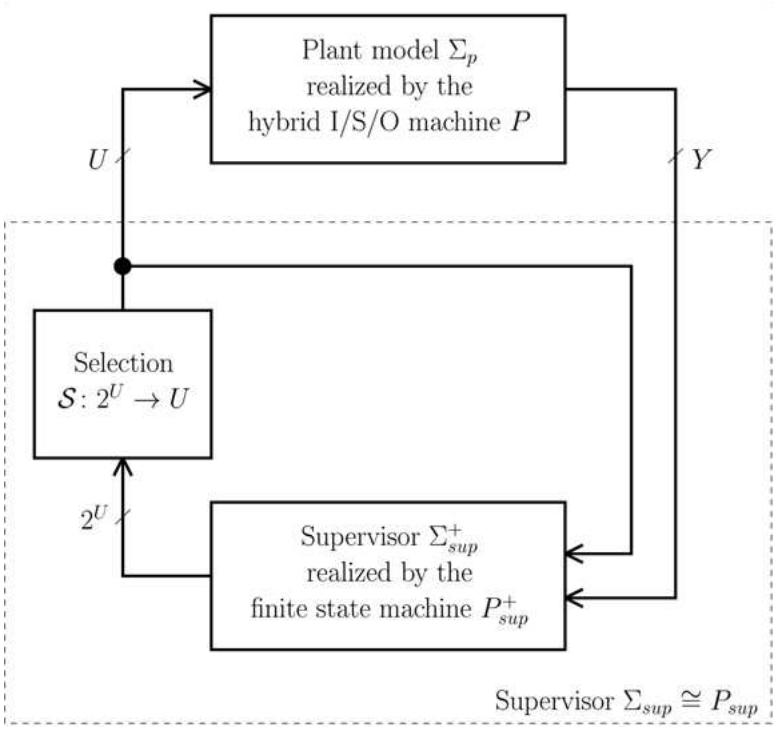

Figure 11. Closed loop scenario, traditional control engineering point of view.

the external signal will be $\omega = (\nu, \mu)$. From Corollary 3, we know that $P \parallel P_{sup}$ is temporally nonblocking. Hence, there exists a transition $(\rho, (\nu, \mu), \rho') \in \lambda_{sup}$ and the closed loop will be in state (ξ', ρ') at time $t + 1$.

Clearly, as this process can be carried on, requirement 1, Section 2, is met. The supervisor never directly affects output symbols. Hence, requirement 2, Section 2, is also met. Finally the closed loop behavior is guaranteed not to violate the specifications.

7. Example

We consider a thermal switched server system consisting of three plates and a radiator, as described in Franke and Moor (1998). 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 drive the temperatures of all plates into a specified range. In Franke and Moor (1998), a rule based switching strategy was found heuristically.

It was formally verified in Moor and Raisch (1998c). As an alternative, we show how our approximation based approach can be used to *formally synthesize* a suitable supervisor.

The following parameters are assumed to be known: the radiator and the environment temperatures $\beta_r \in \mathbb{R}$ and $\beta_e \in \mathbb{R}$, respectively; the corresponding normalized heat transfer coefficients $\alpha_r,\ \alpha_e \in \mathbb{R}^+$; a lower threshold β_-; a middle threshold β_0; an upper threshold β_+.

The evolution of the temperatures w. r. t. continuous time is modeled by linear time invariant differential equations. Let $x_i(\cdot)$ denote the temperature of plate $i \in \{1, 2, 3\}$. While plate i is heated, $x_i(\cdot)$ satisfies

$$\frac{d}{d\tau} x_i(\tau) = \alpha_r\ (\beta_r - x_i(\tau)) + \alpha_e\ (\beta_e - x_i(\tau)) \tag{28}$$

while plate i is not heated, $x_i(\cdot)$ evolves according to

$$\frac{d}{d\tau} x_i(\tau) = 2\ \alpha_e\ (\beta_e - x_i(\tau)) \tag{29}$$

Note that $x_i(\tau) \equiv \beta_m := (\alpha_r\ \beta_r + \alpha_e\ \beta_e)/(\alpha_r + \alpha_e)$ is a stable equilibrium for a heated plate, while $x_i(\tau) \equiv \beta_e$ is a stable equilibrium, if plate i is not heated. We assume $\beta_e < \beta_- < \beta_0 < \beta_+ < \beta_m < \beta_r$.

Whenever a temperature $x_i(\cdot)$ hits a threshold $\nu \in \{\beta_-, \beta_0, \beta_+\}$, the discrete output event

$$(i, \nu) \in Y := \{1, 2, 3\} \times \{\beta_-, \beta_0, \beta_+\} \tag{30}$$

is generated. In response, the supervisor applies a discrete input event

$$\mu \in U = \{1, 2, 3, 4\} \tag{31}$$

where $\mu = 4$ is interpreted as ''radiator off'', while $\mu < 4$ is interpreted as ''radiator positioned at plate μ''.

We look for a supervisor that enforces the following specifications:

a. The temperatures must be driven from an arbitrary initial state $\xi_0 \in [\beta_e, \beta_+]^3$ into the range $(\beta_-, \beta_+]^3$. This warming up procedure must not last longer than t_- external events. Both β_- and t_- are specification parameters.

b. After the warming up procedure, the temperatures are to be kept within $(\beta_-, \beta_+]^3$.

It is obvious, that whenever the upper threshold β_+ is hit, the radiator must be moved to some other plate or switched off. This will keep all temperatures below β_+. Hence, we can regard $X = [\beta_e, \beta_+]^3$ as our state space. The non-trivial part of the above specification is first to drive and then to keep all temperatures above the critical value β_-.

In order to avoid high frequency chattering phenomena, we add the following the requirements:

c. Once the reheating process of a plate has been started, it has to be continued until the plate temperature reaches β_+.

d. Reheating of plate i must not be started unless $x_i(t) \leq \beta_0$.

As we are looking for a purely discrete supervisor, we investigate the plant w. r. t. the discrete time axis defined by the occurrence of output events (logic time). From this point of view, the plant exhibits a discrete external behavior. On the other hand, it is obvious that the evolution of an external trajectory strongly depends on the internal continuous dynamics given by (28) and (29). Furthermore, the continuous valued state $\xi \in X = [\beta_e,\ \beta_+]^3$ at a certain discrete instant of time $t \in \mathbb{N}_0$ together with the current input event $\mu \in U$ uniquely determines the following output event $(i,\ \nu) \in Y$ and the successor state ξ'. Thus, the scenario fits in our framework of hybrid I/S/O machines (Definition 6).

So far, we have set up a plant model and a specification. In order to establish the realization P_l of the l-complete approximation we need to compute the sets of compatible states. While the continuous dynamics w. r. t. continuous time is linear for the above example, the continuous state evolves in a non-linear way w. r. t. discrete (logic) time. Hence, in contrast to linear hybrid automata, the maps f and g from equation (14) turn out to be non-linear. As we cannot perform the iteration (15), (16) exactly, an estimate based

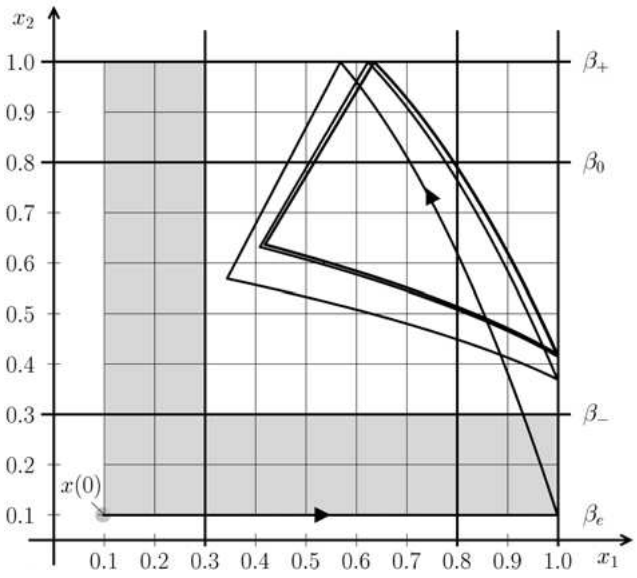

Figure 12. Closed loop simulation.

on state space partitioning and sampling is to be used instead. Here one needs to ensure that (i) no output event is dropped and (ii) the estimate is conservative. For the class of switched linear systems (which includes our example (28) and (29)), we give a detailed computational procedure in Moor and Raisch (1999a).

Here, we state the results for the case where the parameters are chosen as follows: $\beta_e = 0.1$, $\beta_r = 1.5$, $\alpha_e = 0.2$, $\alpha_r = 0.8$, $\beta_+ = 1.0$, $\beta_0 = 0.8$ and $t_- = 6$. For $l \leq 3$, we can only guarantee the specification for the trivial case $\beta_- = \beta_e = 0.1$. For $l = 4$, the accuracy of the l-complete approximation $\Sigma_l \cong P_l$ allows successful supervisory control synthesis up to $\beta_- = 0.275$. For $l = 5$, the accuracy of $\Sigma_l \cong P_l$ allows successful supervisory control synthesis up to $\beta_- = 0.3$. This is illustrated by simulating the model (28), (29) under discrete supervisory control: Figure 12 shows the temperature evolution of the first and second plate where the simulation is started at initial condition (0.1, 0.1, 0.1).

8. Conclusions

In this contribution, we use the framework provided by Willems' behavioral systems theory to suggest an approach for synthesizing supervisory control for hybrid systems. We find the strongest l-complete approximation for a given hybrid system; this approximation can be represented by a finite state machine; hence slightly modified tools from DES theory can be applied to solve the supervisory control problem on the approximation level. It is then shown that the desired closed loop properties are retained if the supervisor is connected to the underlying hybrid system.

References

Alur, R., Courcoubetis, C., Henzinger, T. A., and Ho, P.-H. 1993. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In Grossman et al. *Hybird Systems IV, Lecture Notes in Computer Sciences*, pp. 209–229.

Alur, R., Henzinger, T. A., and Sontag, E. D., editors. 1996. *Hybrid Systems III, Lecture Notes in Computer Sciences*, vol. 1066, Springer-Verlag.

Antsaklis, P. J., Stiver, J. A., and Lemmon, M. 1993. Hybrid system modeling and autonomous control systems. In Grossman et al., *Hybird Systems IV, Lecture Notes in Computer Sciences,* pp. 366–392.

Antsaklis, P. J., Kohn, W., Nerode, A., and Sastry, S., editors. 1995. *Hybrid Systems II, Lecture Notes in Computer Sciences*, vol. 999, Springer-Verlag.

Antsaklis, P. J., Kohn, W., Nerode, A., and Sastry, S., editors. 1997. *Hybrid Systems IV, Lecture Notes in Computer Sciences*, vol. 1273, Springer-Verlag.

Antsaklis, P. J., Kohn, W., Lemmon, M., Nerode, A., and Sastry, S., editors. 1999. *Hybrid Systems V, Lecture Notes in Computer Sciences*, vol. 1567, Springer-Verlag.

Antsaklis, P. J., and Nerode, A., editors. 1998. *IEEE Transactions on Automatic Control*, 43(4), Special issue on hybrid systems.

Chutinan, A., and Krogh, B. H. 1998. Computing polyhedral approximations to flow pipes for dynamic systems. *Proceedings of the 37rd IEEE Conference on Decision and Control*.

Chutinan, A., and Krogh, B. H. 1999. Computing Approximating Automata for a Class of Linear Hybrid Systems. In Antsaklis et al., *Hybird Systems V, Lecture Notes in Computer Sciences*, pp. 16–38.

Cury, J. E. R., Krogh, B. A., and Niinomi, T. 1998. Synthesis of supervisory controllers for hybrid systems based

on approximating automata. In Antsaklis and Nerode, 1998, *IEEE Transactions on Automatic Control,* pp. 564–568.

Dang, T., and Maler, O. 1998. Reachability analysis via face lifting. In Henzinger, T. A. and Sastry, S., editors, *Hybrid Systems: Computation and Control, Lecture Notes in Computer Sciences,* Springer-Verlag, 1386, pp. 96–109,

Evans, R. J., and Savkin, A. V., editors. 1999. *Systems and Control Letters*, 38(3), Special issue on hybrid control systems.

Franke, D., and Moor, T. 1998. Combined rule- and model-based design of a hybrid thermal process. In *Proc. CESA98*, Nabeul-Hammamet, Tunesia, pp. 630–634.

Grossman, R. L., Nerode, A., Ravn, A. P., and Rischel, H., editors. 1993. *Hybrid Systems IV, Lecture Notes in Computer Sciences*, Springer-Verlag, 736.

Klein, E., and Raisch, J. 1998a. Safety enforcement in process control cystems—a batch evaporator example. *Proc. WODES'98—International Workshop on Discrete Event Systems*, IEEE, pp. 327–333.

Klein, E., Kienle, A., and Raisch, J. 1998b. Synthesizing a supervisory control scheme for the start-up procedure of a distillation column—An approach based on approximating continuous dynamics by DES models. *Proc. LSS'98—8th IFAC Colloquium on Large Scale Systems*, pp. 716–721.

Lunze, J. 1995. Stabilization of nonlinear systems by qualitative feedback controllers. *International Journal of Control* 62, 109–128.

Lunze, J. 1994. Qualitative modeling of linear dynamical systems with quantized state measurements. *Automatica* 30, 417–431.

Moor, T. 1998a. Event driven control of switched-integrator-systems. *3rd International Conference ADPM'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, Reims, France, pp. 271–277.

Moor, T., Raisch, J., and O'Young, S. D. 1998b. Supervisory control of hybrid systems via *l*-complete approximations. In Giua, A., Smedinga, R. and Spathopoulos, M., editors, *Proc. WODES'98—International Workshop on Discrete Event Systems*, IEEE, pp. 426–431.

Moor, T., and Raisch, J. 1998c. Estimating reachable states of hybrid systems via *l*-complete approximations. In Bajic, V. B., editor, *Proc. SSCC'98—Advances in Systems, Signals Control and Computers*, IAAMSAD and the South African Branch of the Academy of Nonlinear Sciences, 3, 30–34.

Moor, T., and Raisch, J. 1999a. Discrete control of switched linear systems. *Proceedings of the European Control Conference 1999*.

Moor, T., and Raisch, J. 1999b. Supervisory control of hybrid systems within a behavioral framework. In Evans and Savkin, *Systems and Control Letters*, pp. 157–166.

O'Young, S. D. 1998. Hybrid RTSS. Faculty of Engineering, Memorial University of Newfoundland, Internal Report.

Puri, A., Varaiya, P., and Borkar, V. 1996. Epsilon approximations of differential inclusions. In Alur et al., *Hybird Systems III, Lecture Notes in Computer Sciences,* pp. 362–376.

Raisch, J., and O'Young, S. D. 1997. A totally ordered set of discrete abstractions for a given hybrid or continuous system. In Antsaklis et al., *Hybird Systems IV, Lecture Notes in Computer Sciences*, pp. 342–360.

Raisch, J. 1998a. A hierarchy of discrete abstractions for a given hybrid plant. In *3rd International Conference ADPM'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, Reims, France, pp. 55–62.

Raisch, J., and O'Young, S. D. 1998b. Discrete approximation and supervisory control of continuous systems. In Antsaklis and Nerode, *IEEE Systems on Automatic Control,* pp. 569–573.

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

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

Tittus, M., and Egardt, B. 1994. Control-law synthesis for linear hybrid systems. *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 961–966.

Willems, J. C. 1989. Models for dynamics. *Dynamics Reported* 2, 172–269.

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