Discrete Supervisory Control of Switched Linear Systems

Dieter Franke, Universität der Bundeswehr Hamburg, Thomas Moor, Australian National University, Canberra, and Jörg Raisch, Max-Planck-Institut, Magdeburg

This contribution deals with the synthesis of discrete supervisory control for switched linear systems. Via two successive approximation steps, this hybrid control problem is transformed into a purely discrete problem. The latter can then be solved using a slightly modified version of *P. J. Ramadge*'s and *W. M. Wonham*'s ''supervisory control theory''. By embedding both approximation steps within *J. C. Willems*' ''behavioural systems theory'', we can guarantee that any solution of the resulting discrete synthesis problem will also solve the underlying hybrid problem. All algorithms employed within the approximation and the synthesis procedure terminate after a finite number of steps. The approach is illustrated by means of a simple three-plate thermal system.

1 Introduction

The synthesis of discrete feedback for switched linear systems represents a practically important task in hybrid systems and control theory. As an example, consider a technical plant which is to be operated at different set points. In the vicinity V_i of each set point s_i, it is often possible to adequately describe the plant by a linear model and hence to use local linear feedback control. However, if the plant state ''leaves'' this vicinity, its further evolution is (approximately) modelled by a different set of linear differential equations. We therefore need to switch between a number of suitable linear models to reflect the inherently nonlinear nature of most physical plants. A (higher-level) discrete supervisory controller could then be entrusted the task to force the plant state to remain within V_i despite the occurrence of certain disturbances, or to transfer the plant state in a pre-specified way between different set points s_i. It is precisely this kind of hybrid control problem that we address (and solve) in this paper.

Our approach is as follows: by two successive approximation steps, the hybrid control problem is transformed into a purely discrete problem. The latter can subsequently be solved using a slightly modified version of *P. J. Ramadge*'s and *W. M. Wonham*'s ''supervisory control theory'' [23; 24]. By embedding both approximation steps within *J. C. Willems*' ''behavioural systems theory'' [26; 27], we can *guarantee* that any solution of the resulting discrete synthesis problem will also solve the underlying hybrid problem. Naturally, we need to make sure that all algorithms employed within the approximation and the synthesis procedure terminate after a finite number of steps.

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 [2; 7; 6; 11; 12; 19; 21; 15; 16] are most relevant. One usually distinguishes ''clock time'' scenarios, where the next time instant is determined by the ''ticking'' of an external clock, and ''logic time'' approaches, where the next time instant is determined by certain discrete events being triggered by the continuous component. While [11; 19; 20; 21] are in the first category, the present paper belongs to the more challenging class of ''logic time'' problems. Our paper is broader in scope than other logic-time approaches [12; 6; 25], which restrict the continuous part to consist of integrators only (this is what has come to be known as ''linear hybrid automata'' in the Computer Science literature [1]).

While other papers by the authors [13; 16] represent a thorough investigation of the underlying mathematical problems for a broad class of systems, the present contribution focuses on a specific case and can hence provide specific computational details. It is based on a previous conference contribution [15], but is more tutorial in nature: proofs are mostly omitted and replaced by verbal explanation.

The paper is organized as follows: in Section 2, we discuss a plant model that reflects the above requirements. In particular, it will be shown how the continuous part of the state

variable generates discrete events by crossing certain thresholds. In Section 3, we introduce an approximation procedure that turns the hybrid problem into a purely discrete one. The first approximation step (Section 3.1) is based on a straightforward partitioning of the continuous part of the plant state space. The result of this step is a nondeterministic finite state machine P, which, unfortunately, is not yet amenable to the tools of supervisory control theory. In Section 3.2, we therefore introduce l-complete approximations P_l, $l = 1, 2, \ldots$, for P. We can now apply results from DES (discrete event systems) theory to synthesize a suitable supervisor for P_l (Section 4). If a solution exists, it is guaranteed to work for the underlying continuous (or hybrid) plant model. In Section 5 this is illustrated by means of a simple, though non-trivial, example.

Finally, a remark on notation: signals are represented by small roman letters; if the signal is continuous-valued, the corresponding letter is typed in bold face. Signal spaces (sets) are represented by capital letters, and elements in a signal space by greek letters. For example, $\mathbf{x}$ is a continuous-valued signal taking values in a set X. $\mathbf{x}(t) = \xi \in X$ is the specific value that $\mathbf{x}$ takes at time t.

2 The Plant Model

The plant model consists of a continuous and a discrete part. The former is described by a finite set of linear time-invariant ODEs (ordinary differential equations) in $\mathbb{R}^n$:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mu)\,\mathbf{x}(t) + \mathbf{B}(\mu), \tag{1}$$

where $t \in \mathbb{R}^+$ represents (continuous) time, and $\mathbf{x}(t) \in X \subset \mathbb{R}^n$ is the state at time t. $\mathbf{A}$ and $\mathbf{B}$ are real $n \times n$- and $n \times 1$-matrices, and the values $\mathbf{A}(\mu)$, $\mathbf{B}(\mu)$ are uniquely determined by the last input event $\mu = (\mu_1, \mu_2)$ from a finite set $U = U_1 \times U_2$. Hence, $\mathbf{A}$ and $\mathbf{B}$ remain constant until another input event occurs and causes $\mathbf{A}$, $\mathbf{B}$ to switch to different values. The first component of the input event, μ_1, is generated by the discrete part of the plant model – a finite automaton A – in response to discrete events which are triggered by the continuous signal $\mathbf{x}$ (Figure 1). This implements a state-dependent switching between different evolution laws in (1). The second component, μ_2, is an input that can be "used" by the discrete controller we are about to synthesize. Its measurement information consists of discrete events $\nu \in Y$ emitted by the plant. Like the discrete part of the plant model, the controller can only change its output in response to these events ν. Hence, between the occurrence of measurement events, μ can be treated as a constant parameter of the ODE (1). Note that μ causes the plant model to switch between different evolution laws but cannot make the continuous state variable $\mathbf{x}$ "jump".

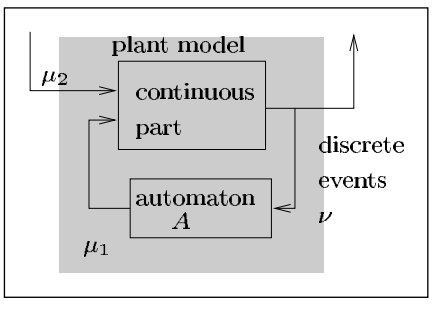

Figure 1: Plant model.

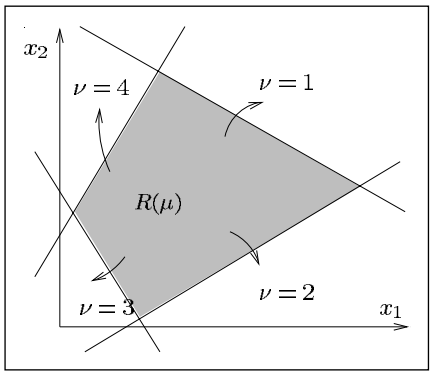

Figure 2: Generation of measurement events.

We now describe how discrete events $\nu \in Y$ are generated: occurrence of an event ν indicates that the continuous state $\mathbf{x}$ is leaving a region $R(\mu) \subset \mathbb{R}^n$ via a certain bounding hyperplane (Figure 2). This can be conveniently formalized by introducing the continuous equation

$$\mathbf{y}(t) = \mathbf{C}(\mu)\mathbf{x}(t) + \mathbf{D}(\mu), \tag{2}$$

where $\mathbf{y}$ is a signal "living" in $\mathbb{R}^p$, and $\mathbf{C}$, $\mathbf{D}$ are real $p \times n$- and $p \times 1$-matrices, respectively. Then, the polyhedron $R(\mu)$ is given by $R(\mu) := \{\xi | \mathbf{C}(\mu)\,\xi + \mathbf{D}(\mu) \geq 0\}$, where "$\geq$" means "componentwise greater or equal", and $\partial R(\mu)$ is the subset of $R(\mu)$ where at least one component of $\mathbf{C}(\mu)\xi + \mathbf{D}(\mu)$ is exactly 0. If the i-th element of $\mathbf{y}$ turns zero, this indicates that the continuous state $\mathbf{x}$ hits the i-th bounding hyperplane and a discrete event $\nu = i$, $i \in \{1, \ldots, p\}$, is generated. We still need to address a few technicalities:

(i) if the state $\mathbf{x}$ hits a vertice of $R(\mu)$, i.e. if two or more components of $\mathbf{y}$ become zero at the same time, ν is taken as the smallest index of these components; this convention is purely to simplify exposition: we could, of course, introduce additional measurement events for these cases; this would imply "finer" measurement information and would hence, in general, lead to a more accurate discrete approximation;

(ii) for the case where an attractive equilibrium in $R(\mu)$ prevents the state to hit $\partial R(\mu)$ for all future, we introduce an additional event $\nu = 0$;

(iii) for any state $\mathbf{x}(t) = \xi$, we have to rule out application of an input μ that selects $R(\mu)$ such that $\xi \notin R(\mu)$. For this case, we introduce the event $\nu = e(rror)$, which has to be prevented by suitable control.

Although this part of the plant model is clearly characterized by the continuous-time evolution of a continuous-valued state variable, its interface with the environment is purely discrete: the environment (consisting of the discrete part of the plant model and the controller we are about to synthesize) observes discrete measurement events $\nu \in Y$

and, in turn, applies discrete input events $\mu \in U$. Thus, the external signal of the continuous part of the plant model is a sequence $(u,\ y) : \mathbb{N}_0 \to U \times Y$ with an input and an output component, both discrete-valued and "living" in discrete-time.

3 Discrete Approximation

Given an arbitrary input $\mu \in U$ and an initial state $\mathbf{x}(0) = \xi \in R(\mu)$, the state $\xi' \in \partial R(\mu)$ at which the next measurement event will be generated is uniquely defined. Still, in general, an explicit representation of ξ' in terms of ξ and μ does not exist – the case $\mathbf{A} \equiv \mathbf{0}$ ("linear hybrid automata") is an exception. As the external signal $(u,\ y)$ is discrete anyway, we can try to circumvent this problem by approximating the continuous part of the plant model by a finite automaton.

In order to apply formal supervisory control synthesis methods on the approximation level, the approximating automaton is required to "cover" all external signals that the continuous part of the plant model deems possible – otherwise the latter could generate a (possibly unacceptable) signal (u, y) which is unknown on the approximation level. Clearly, there is no way to ensure that a supervisor "built" on the approximation level can handle such a situation. *Willems*' "behavioural systems theory" [26; 27] serves as a convenient framework to discuss the desired property of the approximation. There, the external behaviour of a dynamical system is defined as the set of all external signals that are compatible with some set of underlying equations. In our case (we are still examining the continuous part of the plant model), the external behaviour $\mathfrak{B}$ is the set of all signals $(u,\ y) : \mathbb{N}_0 \to U \times Y$ that are compatible with (1) and the "event generating mechanism" described in Section 2. Denoting the behaviour of the approximating automaton by $\mathfrak{B}_{ca}$, we demand $\mathfrak{B} \subseteq \mathfrak{B}_{ca}$, i.e. the approximation is required to be conservative with regard to the external behaviour.

In the following, we describe two approximation steps which are to be carried out successively. The first step is based on the internal structure of the continuous part of the plant model and employs a finite partition of its state space. In the second step, we apply so called *l*-complete approximation [13; 16], which is defined in terms of external behaviours and is therefore not restricted to the specific class of switched linear systems. In spite of the different character of both approximation steps, both are conservative with regard to the external behaviour. This ensures that the overall approximation procedure is also conservative w.r.t. the external behaviour.

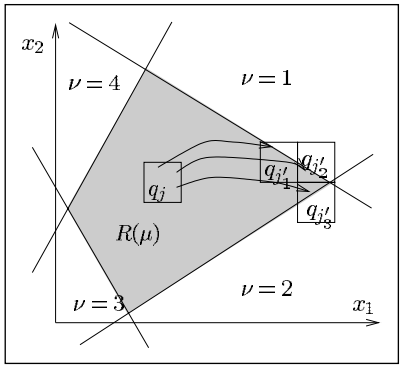

Figure 3: Conservative estimation of next output event and next approximation state.

3.1 Approximation based on partitioning the continuous state space

In the following, we assume for all $\mu \in U$:

(A1) $R(\mu)$ is bounded,

(A2) All eigenvalues of $\mathbf{A}(\mu)$ are within the open left half plane $\mathbb{C}^-$, i.e. $\xi^\star_\mu := -\mathbf{A}^{-1}(\mu)\,\mathbf{B}(\mu)$ is a globally attractive equilibrium.

While (A2) is only meant to facilitate the discussion below, (A1) is essential for our approach: (A1) implies that only the *bounded* subset $\bar{X} := \cup_{\mu \in U} R(\mu) \subset \mathbb{R}^n$ of the continuous state space is relevant. Thus, $\bar{X}$ can be partitioned into a finite number of bounded polyhedra q_j, $j \in J = \{1, \ldots N\}$, i.e. $\bar{X} = \cup_{j \in J} q_j$ and $q_j \cap q_i = \emptyset$ for all $i \neq j$. The finite set J will form the state space of an approximating automaton P.

As we want this approximation to be conservative, we have to address the following problem: assume that the approximation is in state $j \in J$ (i. e. the state $\mathbf{x}(0)$ of the continuous model (1) is an element of q_j), and we apply input μ; construct a superset of the set of possible next events and a superset of the set of possible discrete states j' corresponding to such events (Figure 3). Obviously, "decreasing" the difference between sets and supersets implies increasing approximation accuracy. As we need to *guarantee* that all such events and all such discrete states are covered, forward simulation of a grid of initial points in q_j is not sufficient. Instead, we propose the following strategy:

Step (S1a): To capture all points of a single trajectory, we choose a "fast" sampling rate $1/T$ such that for any $\mathbf{x}(0) = \xi \in q_j$, $\mu \in U$ and any $\tau \in [0,\ T)$ the following implication holds:

$$\mathbf{x}(rT) \in R(\mu) \Longrightarrow \|\mathbf{x}(rT + \tau) - \mathbf{x}(rT)\|_\infty \leq \rho,$$

where $\rho \in \mathbb{R}^+$ is a design parameter, and $\|\ \cdot\ \|_\infty$ denotes the maximum norm, i.e. $\|\xi\|_\infty := \max\{|\xi_i| | 1 \leq i \leq n\}$.

A suitable choice for the sampling interval T has been derived in [15; 17]. Hence, instead of having to check each trajectory for all $t \geq 0$, it suffices to investigate a countable number of points on each trajectory.

Step (S1b): For given $\mu \in U$, define the transition function

$$f_\mu(\xi) := \exp(\mathbf{A}(\mu)\,T)\,\xi + \mathbf{A}(\mu)^{-1}\,(\exp(\mathbf{A}(\mu)\,T) - \mathbf{I}_n)\,\mathbf{B}(\mu)$$

and its extension to sets,

$$f_\mu(q_j) := \{\xi' \mid \xi' = f_\mu(\xi),\ \xi \in q_j\}.$$

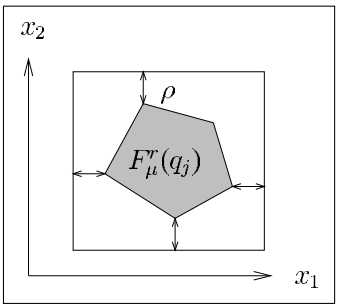

Figure 4: Estimating **x**(t) by boxes.

Both q_j and $R(\mu)$ being polyhedra implies that

$$F_\mu(q_j) := f_\mu(q_j \cap R(\mu)) \cap R(\mu)$$

is a polyhedron. Thus, the r-th iteration $F_\mu^r(q_j)$ is also a polyhedron and can be generated analytically for any finite $r \in \mathbb{N}$. By (S1a) we know that the maximum norm of the continuous state varies by no more than ρ within one sampling interval. Therefore, the continuous state can be safely captured by "boxes" enclosing $F_\mu^r(q_j)$ with "safety distance" ρ (Figure 4):

$$\mathcal{S}(F_\mu^r(q_j), \rho) := \{\xi | \inf_{\xi' \in F_\mu^r(q_j)} |(\xi' - \xi)_i| \leq \rho \forall i\}.$$

Formally, we get as an immediate consequence of (S1a) that for any initial state $\mathbf{x}(0) = \xi \in q_j$ and every t, r such that $r\,T \leq t \leq (r+1)\,T$,

$$\mathbf{x}(\tau) \in R(\mu) \forall \tau \leq t \Longrightarrow \mathbf{x}(t) \in \mathcal{S}(F_\mu^r(q_j), \rho).$$

In other words: as long as the continuous state **x** evolves within $R(\mu)$, it is known to "live" within a sequence of boxes that – once ρ and T have been chosen – depend only on the initial cell, q_j, the value of the discrete input signal, μ, and the sampling index r. Moreover, by assumption (A2), only a finite number $r^+ \in \mathbb{N}$ of sampling instants need to be considered: if $\xi_\mu^\star \notin R(\mu)$, the polyhedron $F_\mu^r(q_j)$ becomes empty for some $r = r^+ \in \mathbb{N}$; see Figure 5. If $\xi_\mu^\star \in R(\mu)$, using Lyapunov-type arguments (see [17]), an invariant region of attraction $G(\xi_\mu^\star)$ can be constructed within the box $\mathcal{S}(\{\xi_\mu^\star\}, 2\rho)$. Then, for some $r = r^+ \in \mathbb{N}$, the polyhedron $F_\mu^r(q_j)$ lies within $G(\xi_\mu^\star)$ and any future evolution of **x** is conservatively approximated by the single box $\mathcal{S}(\{\xi_\mu^\star\}, 2\rho)$; see Figure 6. In both cases, the iteration $F_\mu^r(q_j)$ can be faithfully terminated for $r = r^+$. We are now able to trace an infinite number of trajectories in continuous time by checking the evolution of a single box for a finite number of sampling instants. This sets the stage for the remaining step.

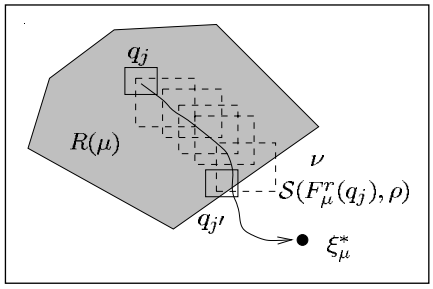

Figure 5: Equilibrium $\xi_\mu^\star$ outside $R(\mu)$.

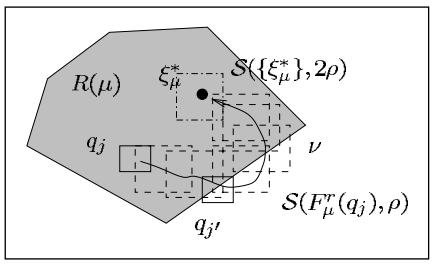

Figure 6: Equilibrium $\xi_\mu^\star$ within $R(\mu)$.

Step (S1c): We are now in a position to answer the question posed at the beginning of this section: assume that $\mathbf{x}(0) = \xi \in q_j$ and the input event μ has been applied; which measurement events ν can occur next, and in which cell $q_{j'}$ could the continuous state then be? To put it more formally: we ask whether $(j, (\mu, \nu), j')$ is to be a transition in our approximating automaton. We distinguish four cases:

case (i) $\nu \in \{1, \ldots, p\}$ (i.e. ν is a "regular event") and $\xi_\mu^\star \notin R(\mu)$. This situation is depicted in Figure 5: clearly, if a trajectory starting in q_j is to leave $R(\mu)$ via the ν-th bounding hyperplane and the cell $q_{j'}$, both of the following conditions need to hold (for a formal proof, see [17; 15]):

(C1) $\min\{(\mathbf{C}(\mu)\,(\mathbf{A}(\mu)\,\xi + \mathbf{B}(\mu)))_\nu \mid \xi \in \partial R_\nu(\mu) \cap q_{j'}\} < 0,$

(C2) $[\cup_{r \leq r^+} \mathcal{S}(F_\mu^r(q_j), \rho)] \cap \partial R_\nu(\mu) \cap q_{j'} \neq \emptyset.$

case (ii) $\nu \in \{1, \ldots, p\}$ and $\xi_\mu^\star \in R(\mu)$. This situation is illustrated in Figure 6. Condition (C1) carries over from case (i) and is therefore still necessary. Minor modifications to (C2) in order to consider the final box around the equilibrium give:

(C3) $[\cup_{r \leq r^+} \mathcal{S}(F_\mu^r(q_j),\ \rho) \cup \mathcal{S}(\{\xi_\mu^\star\},\ 2\,\rho)] \cap\ \partial R_\nu(\mu)\ \cap\ q_{j'} \neq \emptyset.$

case (iii) $\nu = 0$ is the additional event signifying that an attractive equilibrium $\xi_\mu^\star \in R(\mu)$ prevents the trajectory from escaping $R(\mu)$. A necessary condition for this is that $F_\mu^r(q_j) \neq \emptyset$ for all $r \in \mathbb{N}$. Recall that the iteration in r is terminated if either $F_\mu^{r^+}(q_j)$ "moves" inside an invariant domain of attraction or $F_\mu^{r^+}(q_j)$ becomes empty. Recall also that, by definition of $F_\mu(q_j)$, the set $F_\mu^{r^+}(q_j)$ is empty whenever $F_\mu^r(q_j) = \emptyset$ for some $r \leq r^+$. Hence, in terms of the suggested iteration, a necessary condition can also be formulated as

(C4) $F_\mu^{r^+}(q_j) \neq \emptyset.$

case (iv) $\nu = e$ represents the case where $\mathbf{x}(0) = \xi \notin R(\mu)$. Obviously, for this to happen, it is necessary that

(C5) $\{\xi \mid \xi \in q_j,\ \xi \notin R(\mu)\} \neq \emptyset.$

This completes the construction of a finite automaton P that approximates the continuous part of the plant model. Recall that each step in the proposed procedure is finite. Cases (i) to (iv) can be summarized as follows:

$(j, (\mu, \nu), j')$ is defined to be a transition in P if and only if either

$$\begin{aligned} & \nu \in \{1, \ldots p\}, \xi^{\star}_{\mu} \notin R(\mu), \text{(C1) and (C2)}, \\ \text{or} \quad & \nu \in \{1, \ldots p\}, \xi^{\star}_{\mu} \in R(\mu), \text{(C1) and (C3)}, \\ \text{or} \quad & \nu = 0, j = j' \text{ and (C4)}, \\ \text{or} \quad & \nu = e, j = j' \text{ and (C5)}. \end{aligned}$$

Clearly, as only *necessary* conditions are involved, the automaton P forms a conservative approximation in the following sense: whenever applying the input event μ to the initial state $\mathbf{x}(0) = \xi \in q_j$ takes the continuous part of the plant model into state $\xi' \in q_{j'}$ by the time the next measurement event ν is generated, $(j, (\mu, \nu), j')$ will be a transition in P. As a consequence, P realizes a conservative approximation of the continuous part of the plant model w.r.t. the external behaviour.

Now, the discrete part of the plant model needs to be taken into account. For this purpose, we interconnect the automaton A (see Figure 1) and the approximating automaton P. This is done by means of synchronizing the common input component $\mu_1 \in U_1$ and the measurement events $\nu \in Y$. Standard procedures from the field of DES are available for "building" $P \parallel A$. Clearly, $P \parallel A$ realizes a conservative approximation of the underlying hybrid plant model shown in Figure 1. For notational simplicity, $P \parallel A$ will henceforth be referred to as P, its external behaviour as $\mathfrak{B}_{ca}$. Furthermore, $\mathfrak{B}$ will from now on denote the external behaviour of the overall hybrid plant model.

3.2 Generating an "instantly observable" automaton by *l*-complete approximation

Supervisor synthesis can be based most effectively on "instantly observable" finite automata. They are characterized by the fact that, at any instant of time, the internal state is uniquely determined by the sequence of all previously observed external inputs and measurements. Hence, there exists an observer that provides the exact value of the state without any delay. The property of "instant observability" is equivalent to determinism within *Ramadge*'s and *Wonham*'s supervisory control theory [23; 24]; it is also equivalent to the notion of "past inducedness" within *Willems*' behavioural framework [26; 27].

The finite automaton P derived in Section 3.1 is *not* instantly observable: in general, the measurement information is not sufficiently rich to uniquely determine the discrete state $j \in J$. This is intuitively clear from Figure 3. In theory, any finite automaton can be turned into an instantly observable automaton without affecting its external behaviour by simply re-encoding its state space. Unfortunately, the size of the new state space depends exponentially on the size $|J|$ of J, an upper bound being given by $2^{|J|}$. As we want our partition of the continuous state space to be reasonably fine, a "large" J is to be expected. In a purely continuous example with three dimensional state space, we used a partition with 10 quantization intervals per axis, resulting in $|J| = 10^3$ discrete states. Clearly, transforming an automaton of this size into an instantly observable automaton is in general not feasible.

For this reason, we subject P to another approximation step, which is based on the notion of *l-completeness*. Roughly speaking, the property of l-completeness, $l = 1, 2, \ldots$, limits the memory of a dynamical system to the l most recent input events and the l most recent measurement events; a formal definition of l-completeness in terms of the external behaviour can be found in [26; 27]. Any l-complete external behaviour $\mathfrak{B}_l$ can be realized by means of a state model with the state variable memorizing the l most recent external events. Clearly, such a realization will be instantly observable. As in our particular scenario the external signal "lives" on a finite set, $U_2 \times Y$, the state variable of such a realization is also restricted to a finite number of different values, an upper bound being given by $(|U_2|\,|Y|)^{(l+1)}$.

In general, one cannot expect the external behaviour $\mathfrak{B}_{ca}$ of the approximating automaton P to be l-complete. We therefore look for the *strongest l-complete approximation* of $\mathfrak{B}_{ca}$. This is defined to be the smallest superset $\mathfrak{B}_l \supseteq \mathfrak{B}_{ca}$ which is l-complete [13; 16]. In other words: $\mathfrak{B}_l$ is the most accurate conservative approximation of $\mathfrak{B}_{ca}$ that is l-complete and thus – recall the discussion above – realizable by an instantly observable finite automaton P_l. The bound $l \in \mathbb{N}$ on the memory of this approximation can be interpreted as a design parameter: its effect on approximation accuracy is monotone in the sense of $\mathfrak{B}_l \supseteq \mathfrak{B}_{l'} \supseteq \mathfrak{B}_{ca}$ for all $l' \geq l$. Thus, when increasing l, one expects to gain accuracy. [13; 16; 17] provide an extensive discussion of l-complete approximations within the behavioural framework, addressing a general class of hybrid systems. A most significant aspect for our scenario (with $\mathfrak{B}_{ca}$ being realized by a finite state machine P) is that computation of P_l can be implemented in a straightforward manner; see [13], Proposition 2, Theorem 2.

We now summarize the overall approximation procedure: in a first step, based on sampling and partitioning of the continuous state space, we construct a finite automaton which conservatively approximates the continuous part of the plant model. This is connected to the discrete part of the plant model – another finite automaton – to give a conservative discrete approximation of the overall plant. In a second step, l-complete approximation is applied yielding an instantly observable finite automaton P_l. As both approximation steps are conservative with regard to the external behaviour, this is also true for the entire approximation procedure: P_l provides all external signals that are compatible with the underlying hybrid plant model.

4 Supervisory Control

The plant described in Section 2 is to be connected to some supervisory controller by means of input events $\mu_2 \in U_2$ and measurement events $\nu \in Y$; see Figure 1. In response to any measurement event ν "seen" by the controller, it ap-

plies an input event μ_2 to the plant. While the plant internally exhibits both discrete and continuous dynamics, we restrict attention to the case of purely discrete supervisory controllers, which are to be realized by finite automata P_{sup}.

A typical control task (specification) is to prevent certain measurement events which correspond to undesired situations, e. g. the continuous part of the plant state evolving into a "forbidden" region. More sophisticated specifications also incorporate dynamcis, implying that the occurrence of certain events affects what needs to be prevented in future; an example are specifications that aim at forcing the plant into some cyclic behaviour. In order to capture various types of control goals, we formalize specifications by a set of acceptable external signals; it is assumed that this set of acceptable signals can be realized by a finite automaton P_{spec}, i.e. the external signals exhibited by P_{spec} are precisely those which are acceptable.

If both plant and specification are given, the problem of supervisory control synthesis is to construct a supervisor that forces the plant to evolve on acceptable trajectories only. This problem is addressed by *Ramadge*'s and *Wonham*'s "supervisory control theory" [23; 24]. From the application point of view, a particularly important result from [23; 24] is a terminating algorithm which either solves the problem in an optimal (i.e. minimally restrictive) way, and hence constructs a suitable supervisor, or tells the user that no solution exists. As the algorithm requires the plant to be realized by an instantly observable finite automaton, it can neither be applied to the switched linear system described in Section 2 nor to the approximation P derived in Section 3.1. On the other hand, it may – at least in principal– be applied to the approximation P_l discussed in Section 3.2.

Then, of course, the following question arises immediately: if we find a suitable supervisor P_{sup} for the approximation P_l, will we be able to *guarantee* that this supervisor also enforces the specifications when connected to the underlying hybrid plant model? In Section 3, we have already pointed out that to give an affirmative answer, P_l is required to be a conservative approximation w.r.t. the external behaviour. Within *Willems*' behavioural framework, the line of thought is as follows: if a supervisor solves the problem on the approximation level, it is capable of restricting the external behaviour of the approximation, $\mathfrak{B}_l$, to the set $\mathfrak{B}_{spec}$ of acceptable trajectories (the latter being realized by the automaton P_{spec}); as the external behaviour of the underlying hybrid system, $\mathfrak{B}$, only holds trajectories which are also contained in $\mathfrak{B}_l$, the supervisor will prevent any trajectory in $\mathfrak{B}$ from "escaping" the specification behaviour $\mathfrak{B}_{spec}$. A formal proof requires a thorough investigation of supervisory control theory within the behavioural framework (see [16; 17] for details). We also point out that, due to the specific input/output structure of our system class, some minor modifications to *Ramadge*'s and *Wonham*'s theory become necessary; this is extensively discussed in [13; 16]. A suitable supervisory controller synthesis procedure has been coded in C++ with an object oriented architecture [18].

In summary, applying the results from [13; 16; 17] to the class of switched linear systems from Section 2 and the conservative, instantly observable approximation described in Section 3, we get the desired statement: *a supervisory controller synthesized on the basis of approximation P_l will indeed solve the synthesis problem for the underlying switched linear system*.

5 Example: a Three-Plate Thermal System

We consider a thermal switched server system consisting of three plates and a radiator, as described in [9]. 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 [9], a rule based switching strategy was found heuristically. It was formally verified in [14]. 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 α_r, $\alpha_e \in \mathbb{R}^+$; a lower threshold β_-; a middle threshold β_0; an upper threshold β_+.

The evolution of the temperatures in continuous time is modelled by linear time-invariant differential equations. Let $\mathbf{x}_i$ denote the temperature of plate $i \in \{1, 2, 3\}$. While plate i is heated, $\mathbf{x}_i$ satisfies

$$\dot{\mathbf{x}}_i(t) = \alpha_r\,(\beta_r - \mathbf{x}_i(t)) + \alpha_e\,(\beta_e - \mathbf{x}_i(t)); \tag{3}$$

while plate i is not heated, $\mathbf{x}_i$ evolves according to

$$\dot{\mathbf{x}}_i(t) = 2\,\alpha_e\,(\beta_e - \mathbf{x}_i(t)). \tag{4}$$

Note that $\mathbf{x}_i \equiv \beta_m := (\alpha_r\,\beta_r + \alpha_e\,\beta_e)/(\alpha_r + \alpha_e)$ is a stable equilibrium for a heated plate, while $\mathbf{x}_i \equiv \beta_e$ 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 $\mathbf{x}_i$ hits one of the thresholds β_-, β_0 or β_+, this shall be notified by a measurement event. In response to any of these events, the supervisor applies a discrete input event

$$\mu_2 \in U_2 = \{1, 2, 3, 4\},$$

where $\mu_2 = 4$ is interpreted as "radiator off", while $\mu_2 < 4$ is interpreted as "radiator positioned at plate μ_2".

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 warm-up procedure must not last longer than t_- external events. Both β_- and t_- are specification parameters.

(b) After the warm-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 prevent all temperatures from rising above β_+. Hence, we can regard $\bar{X} = [\beta_e, \beta_+]^3$ as the relevant part of the state space. The non-trivial part of the 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 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 $\mathbf{x}_i(t) \leq \beta_0$.

We now briefly explain that the considered scenario fits into the framework of switched linear system as provided in Section 2. How to set up the system matrices $\mathbf{A}(\mu)$ and $\mathbf{B}(\mu)$ is obvious from (3) and (4). The regions $R(\mu)$ are boxes with vertices determined by the threshold values β_-, β_0, and β_+. This defines the matrices $\mathbf{C}(\mu)$ and $\mathbf{D}(\mu)$. The next (in all likelihood painful) task would be to specify the automaton A in order to select the regions $R(\mu)$ such that the continuous state is always contained in $R(\mu)$. Instead, we extend U_2 such that the boxes $R(\mu)$ are exclusively selected by the supervisor: $U = U_2$, and the automaton A can be dispensed with. The selection process now becomes part of the supervisory control synthesis problem and hence will be addressed by the synthesis procedure (recall the discussion in Section 3.1). Instead of indices $\nu \in Y$ (representing bounding hyperplanes of boxes $R(\mu)$), we can now use a set of measurement events which are much easier to interprete: each event

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

denotes that threshold β is being hit by the temperature of plate i. Then, the informal specifications (a) to (d) can be easily turned into a specification automaton P_{spec}.

We can now apply our two approximation steps followed by the synthesis procedure outlined in Section 4. If a suitable supervisor exists on the approximation level, it will be found by the synthesis procedure, and it is guaranteed to solve the control problem for the underlying switched linear system. If no suitable supervisor exists on the approximation level, this will be notified by the synthesis procedure. It is intuitively clear, that to meet stronger specifications, a finer approximation is needed. Thus, if no suitable supervisor is found for a particular approximation, we may either relax the specifications, or increase approximation accuracy by choosing a finer state space partition and/or by increasing the parameter l.

We now report 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 $t_- = 6$. Partitioning of the continuous state space $\bar{X}$ is defined by independent quantization of all three temperatures, with quantization intervals of length 0.025. This results in 36^3 partition cells. The parameters l and β_- are varied in order to highlight the relation between approximation accuracy and achievable specifications: for $l \leq 3$, we can only guarantee the specifications for the trivial case $\beta_- = \beta_e = 0.1$. For $l = 4$, successful supervisory control synthesis is possible up to $\beta_- = 0.275$. For $l = 5$, we can enforce the specifications for values of β_- up to 0.3. This is illustrated by simulating the model (3), (4) under discrete supervisory control: Figure 7 shows the temperature evolution of the first and second plate for the initial condition (0.1, 0.1, 0.1).

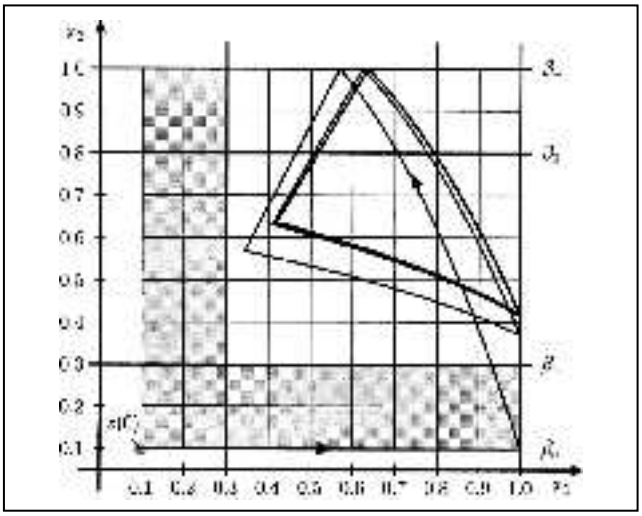

Figure 7: Closed loop simulation.

6 Conclusions

In this contribution, we suggest an approach for synthesizing supervisory control for switched linear systems based on two conservative approximation steps. First, the switched linear system is approximated by sampling and state space partitioning. This results in a finite state machine P, which, in general, is not instantly observable. In a second step, we compute the strongest l-complete approximation P_l of P. By construction, P_l is an instantly observable finite automaton. We can now apply slightly modified tools from DES theory to solve the supervisory control problem on the approximation level. The desired closed loop properties are retained if the supervisor is connected to the underlying switched linear system. All algorithms employed within the approximation and the synthesis procedure terminate after a finite number of steps. The approach is illustrated by means of a simple three-plate thermal system.

Acknowledgement

Support from Deutsche Forschungsgemeinschaft under grants Fr 598/6-3 and Ra 516/3-1 is gratefully acknowledged.

References

[1] 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 [10], pp. 209–229.

[2] Antsaklis, P. J., Stiver, J. A. and Lemmon, M. 1993. Hybrid system modelling and autonomous control systems. In [10], pp. 366-392,

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

[4] 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.

[5] Antsaklis, P. J. and Nerode, A., editors. 1998. *IEEE Transactions on Automatic Control*, vol 43, no 4, Special issue on hybrid systems.

[6] Chutinan, A. and Krogh, B. H. 1999. Computing Approximating Automata for a Class of Linear Hybrid Systems. In [4], pp. 16–38.

[7] Cury, J. E. R., Krogh, B. A. and Niinomi, T. 1998. Synthesis of supervisory controllers for hybrid systems based on approximating automata. In [5], pp. 564-568.

[8] Evans, R. J. and Savkin, A. V., editors. 1999. *Systems and Control Letters*, vol 38, no 3, Special issue on hybrid control systems.

[9] 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.

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

[11] Lunze, J. 1995. Stabilization of nonlinear systems by qualitative feedback controllers. In *International Journal of Control*, vol. 62, pp. 109–128.

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

[13] 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*, IEE, pp. 426–431.

[14] 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, vol. 3, pp. 30–34.

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

[16] Moor, T. and Raisch, J. 1999b. Supervisory control of hybrid systems within a behavioural framework. In [8], pp. 157-166.

[17] Moor, T. 2000. *Approximationsbasierter Entwurf diskreter Steuerungen für gemischtwertige Regelstrecken*, Forschungsberichte aus dem Max-Planck-Institut für Dynamik komplexer technischer Systeme, Vol. 2, Shaker-Verlag, Aachen, 2000. Also PhD thesis, Fachbereich Elektrotechnik, Universität der Bundeswehr Hamburg.

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

[19] Raisch, J. and O'Young, S. D. 1997. A totally ordered set of discrete abstractions for a given hybrid or continuous system. In [3], pp. 342–360.

[20] 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. Extended version published as [22].

[21] Raisch, J. and O'Young, S. D. 1998b. Discrete approximation and supervisory control of continuous systems. In [5]. pp. 569–573.

[22] Raisch, J. 1999. A hierarchy of discrete abstractions for a hybrid plant. In *Journal Européen des Systèmes Automatisés*, vol. 32, pp. 1073–1095. Special issue ADPM98.

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

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

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

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

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

Manuscript received: 15th February 2000.

Prof. Dr.-Ing. Dieter Franke is head of the control group, department of electrical engineering, at the University of the Federal Armed Forces, Hamburg. His main research interests are nonlinear control systems, discrete-event and hybrid systems, and distributed parameter systems.

Address: Universität der Bundeswehr Hamburg, D-22039 Hamburg, Fed. Rep. of Germany, Tel.: +49 40 6541 2754,
E-Mail: dieter.franke@unibw-hamburg.de

Dr.-Ing. Thomas Moor is a research fellow in the Research School of Information Sciences and Engineering at the Australian National University. His main research interests are in the areas of hybrid systems and discrete event systems.

Address: Research School of Information Sciences and Engineering, Australian National University, Canberra ACT 0200, Australia, Tel.: +61 2 6279 8680, E-Mail: thomas.moor@anu.edu.au

Priv.-Doz. Dr.-Ing. Jörg Raisch leads a research group on systems and control theory at the recently founded *Max-Planck-Institut für Dynamik komplexer technischer Systeme*, Magdeburg. His main research interests are in the areas of hybrid systems, multivariable control, process control, and hierarchical control.

Address: Max-Planck-Institut für Dynamik komplexer technischer Systeme, Leipziger Str. 44, 39120 Magdeburg, Fed. Rep. of Germany, Tel.: +49 391 6117-523, E-Mail: raisch@mpi-magdeburg.mpg.de