DISCRETE CONTROL OF SWITCHED LINEAR SYSTEMS

T. Moor[*]**, J. Raisch**[**]

[*]Universität der Bundeswehr Hamburg, D-22039 Hamburg, FR Germany
fax: +49 40 6541 2822, e-mail: thomas.moor@unibw-hamburg.de

[**]Max-Planck-Institut für Dynamik komplexer technischer Systeme, D-39120 Magdeburg, FR Germany
fax: +49 391 6117 501, e-mail: raisch@mpi-magdeburg.mpg.de

Keywords: Switched continuous systems, hybrid systems, conservative approximations, l-complete approximations, supervisory control.

Abstract

Switched linear systems exhibit a continuous state evolving along the continuous flow of time according to linear time invariant differential equations. Furthermore, a discrete interface to the environment is provided, acting on input signals by switching between a finite number of differential equations and generating output signals when the continuous state crosses certain boundaries. We suggest a conservative approximation scheme based on sampling, state partitioning and l-completion realized by a finite past induced state machine. The control problem is investigated on the approximation level. If a solution exists, it also solves the problem for the switched linear system.

1 Introduction

In [mo98a], we suggested an approach for synthesizing supervisory control for a general class of hybrid systems. It is based on the framework provided by *Willems'* behavioural systems theory (e. g. see [wi91]), and *Ramadge*'s and *Wonham*'s supervisory control theory (e. g. [ram89]). While [mo98a] addresses the theoretical aspects, the current contribution applies our approach to the more specific scenario of switched linear systems.

The continuous dynamics of a switched linear system is represented by a finite set of linear time invariant state space systems. A discrete input is used to switch between the linear systems. The input also selects a polyhedron serving as a boundary for the state trajectory. An output signal is generated, whenever the continuous state (evolving along continuous flow of time) is about to cross the boundary of the polyhedron. We demand the input to be constant as long as no output event occurs, since we are going to close the loop by a discrete supervisor. Thus, while a switched linear system internally involves continuous dynamics, it induces a discrete external behaviour. A switched linear system can be modelled by a hybrid automaton, but the latter, in general, will *not* be linear (see Remark 1). Hence, neither control synthesis nor closed loop analysis can be performed by established methods (e. g. [tit94], [al93]): supervisory control synthesis for switched linear systems is a non trivial problem.

This paper is organized as follows: in Section 2, we give a definition of state machines, related terminology and some basic properties. A detailed description of the switched linear system scenario is provided in Section 3. Section 4 describes a conservative past induced finite state approximation based on sampling, partitioning, and l-completion as introduced in [mo98a]. In Section 5 we summarize the main result of [mo98a], synthesizing a supervisor solving the control problem for a given switched linear system. Finally, Section 6 provides an example.

2 State machines

State machines are a common framework when modelling discrete time dynamics; the purpose of this section is to collect basic definitions for the reader's convenience.

Definition 1. Let the sets W, 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. The tuple $P = (X,\, W,\, \delta,\, X_0)$ is called a *state machine*. If $|W| \in \mathbb{N}$ and $|X| \in \mathbb{N}$ (both sets are finite), P is said to be a *finite state machine*. The behaviour $\mathfrak{B}_s := \{(w,x)|\ (x(t),\, w(t),\, x(t+1)) \in \delta\ \forall\ t \in \mathbb{N}_0,\ x(0) \in X_0\}$ is referred to as the induced *full behaviour*. The *external behaviour* $\mathfrak{B}$ induced by P is defined to be the projection of $\mathfrak{B}_s$ onto $W^{\mathbb{N}_0}$, that is $\mathfrak{B}_{ex} := \mathcal{P}_W\, \mathfrak{B}_s := \{w|\ \exists\, x :\ (w,x) \in \mathfrak{B}_s\}$. A state machine P' with induced external behaviour $\mathfrak{B}$ is said to be a *realization* of $\mathfrak{B}$. This is denoted by $\mathfrak{B} \cong P'$. □

The induced external behaviour $\mathfrak{B}$ is the set of all trajectories which are compatible with the state machine from the environment point of view. As we use state machines to model a phenomenon we want to control, the supervisor will form the environment and relies only on $\mathfrak{B}$. Hence, the external behaviour $\mathfrak{B}$ plays an important role in supervisor synthesis and *Willems*' "behavioural approach" forms a natural framework for our investigations; see [wi91] for an overview. However, for the scope of this paper, we stick to the realization level as far as possible: all behaviours we deal with will be realized by state machines.

In the sequel, we introduce some rather basic terminology related to state machines:

Definition 2. 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 (elements in the next state relation) from α connecting a_0 with a_1. The state machine P_a is said to be *reachable* if every state $a_1 \in A$ is reachable. *Nonblocking:* The state machine P_a is called *nonblocking*, if for every reachable state $a \in A$ there exists $w \in W$, $a' \in A$ such that $(a,\, w,\, a') \in \alpha$. *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 $(b,\, w,\, b') \in \beta$. □

When a state machine P realizes the behaviour $\mathfrak{B}$, a reachable state machine P' realizing $\mathfrak{B}$ can be constructed by removing unreachable states and related transitions. In analogy, a nonblocking reachable state machine P'' realizing $\mathfrak{B}$ can be constructed. If the state space of P is finite, both procedures are finite, and the state spaces of P' and P'' are finite.

Definition 3. (See [wi89], section 2.2.1) Let $P = (X,\, W,\, \delta,\, X_0)$ denote a state machine with induced state behaviour $\mathfrak{B}_s$. Then P is said to be *past induced*, if $t \in \mathbb{N}_0$, $(w',\, x')$, $(w'',\, x'') \in \mathfrak{B}_s$, $w'|_{[0,t)} = w''|_{[0,t)}$ implies $x'(t) = x''(t)$. □

Here, $w|_{[t_1,t_2)}$ denotes the restriction of the map $w\colon \mathbb{N}_0 \to W$ to the domain $[t_1, t_2) \cap \mathbb{N}_0$. To keep notation reasonably compact, $w|_\emptyset = w'|_\emptyset$ by definition holds for all maps w, w'.

A past induced realization is "instantaneously state observable": at every moment $t \in \mathbb{N}_0$, we can figure out the state $x(t)$ by investigating only past external signals $w|_{[0,t)}$. Thus, past inducedness is a crucial property when investigating control related tasks: the synthesis method suggested in Section 5 relies on a past induced realization. A nonblocking realization P is past induced if and only if (i) the set of initial conditions X_0 holds only one element and (ii) for any reachable state ξ and any external signal ω there exists one unique state ξ' such that $(\xi,\, \omega,\, \xi') \in \delta$. Again, we build a nonblocking past induced state machine $P_{past} = (Q,\, W,\, \lambda,\, Q_0)$ from any given state machine P without affecting the induced external behaviour. Here, the construction can be done based on $Q = 2^X$ and $Q_0 = \{X_0\}$; hence, if P is finite, P_{past} can be chosen to be finite too. However, this is likely to result in a state space of a size we cannot handle.

For the purpose of supervisory control, we need to distinguish between external signals which can be controlled (inputs) and such which cannot (outputs). To keep notation as simple as possible, throughout this paper only one product decomposition $W = U \times Y$ is considered, where U denotes the set of input signals, and Y the set of output signals. By $(u,\, y) = w \in W^{\mathbb{N}_0}$ we always refer to this decomposition.

Definition 4. A state machine $P = (X,\, W,\, \delta,\, X_0)$, $W = U \times Y$, is said to be an *I/S/- machine*, if for every reachable $\xi \in X$, $\mu \in U$, there exists a $\nu \in Y$, $\xi' \in X$ such that $(\xi,\, (\mu,\, \nu),\, \xi') \in \delta$. □

Note that I/S/- machines are nonblocking. Unlike *Willems*' I/S/O systems, we do not demand the output to process the input (see [wi91], Def. VIII.2, IX.1).

3 Switched linear systems

In this section, a system class is introduced where the external behaviour is discrete, while the internal dynamics are represented by a finite number of linear time invariant differential equations:

$$\mu \in U := \{1,\, \dots\, m\}\,, \tag{1}$$

$$A(\mu) \in \mathbb{R}^{n\times n},\, B(\mu) \in \mathbb{R}^n\,, \tag{2}$$

$$C(\mu) \in \mathbb{R}^{p\times n},\, D(\mu) \in \mathbb{R}^p\,, \tag{3}$$

$$\dot{x}^c(\tau) = f(x^c(\tau),\, \mu) := A(\mu)\, x^c(\tau) + B(\mu)\,, \tag{4}$$

$$y^c(\tau) = g(x^c(\tau),\, \mu) := D(\mu)\, x^c(\tau) + C(\mu)\,. \tag{5}$$

Here, a discrete input signal $\mu \in U$ implements selecting the right hand side $f(\,\cdot\,,\, \mu)$ and the output map $g(\,\cdot\,,\, \mu)$. For any given initial state $x^c(0) = x_0 \in \mathbb{R}^n$ and any constant $\mu \in U$, the differential equation (4) has a unique solution on the continuous time axis $\mathbb{R}_0^+$ denoted by

$$\varphi(\,\cdot\,,\, x_0,\, \mu)\colon \mathbb{R}_0^+ \to \mathbb{R}^n\,. \tag{6}$$

We now discuss how discrete output signals are generated. Let an initial state $x_0 \in R(\mu)$ be given, where $R(\mu) \subseteq \mathbb{R}^n$ denotes the polyhedron defined as the set of states such that all components of the continuous output are non-negative. Then, as soon as the state hits the boundary of the polyhedron $R(\mu)$ (denoted by $\partial R(\mu)$), an output signal is generated. The situation is characterized by a component of the continuous output being about to become negative. The index of that component is taken as the output signal $y_1 \in \{1,\, \dots\, p\}$. If this index is not determined uniquely (e. g. when the state hits a vertice of $R(\mu)$), an arbitrary index of an output component becoming negative is taken as y_1. If either x_0 happens not to be within $R(\mu)$, or an attractive equilibrium in $R(\mu)$ prevents the state to hit $\partial R(\mu)$ for all future, this is treated as an "error": applying the input μ to such an initial state x_0 is "forbidden". Since we want to form an I/S/- machine, we introduce the dummy output 0 for this situation. One of the control goals will be that the supervisor disables this kind of forbidden inputs. Given $x_0 \in \mathbb{R}^n$ and $\mu \in U$, the generation of discrete output signals is formalized by:

$$Y := \{0,\, \dots\, p\}\,, \tag{7}$$

$$R(\mu) := \{\xi |\, g(\xi,\, \mu) \geq 0\}\,, \tag{8}$$

$$\tau_1(x_0,\, \mu) := \sup\{\tau |\, \varphi(\tau',\, x_0,\, \mu) \in R(\mu)\; \forall\; 0 \leq \tau' < \tau\} \tag{9}$$

and, if $0 < \tau_1(x_0,\, \mu) < \infty$

$$x_1(x_0,\, \mu) := \varphi(\tau_1(x_0,\, \mu),\, x_0,\, \mu)\,, \tag{10}$$

$$Y_1(x_0,\, \mu) := \{i |\, e_i^\top\, g(x_1(x_0,\, \mu),\, \mu) = 0\}\,, \tag{11}$$

or, if $\tau_1(x_0,\, \mu) \in \{0,\, \infty\}$

$$x_1(x_0,\, \mu) := x_0,\, Y_1(x_0,\, \mu) := \{0\}\,. \tag{12}$$

Here, e_i is the i-th unit vector. By equations (6) to (12) an I/S/-machine $P := (\mathbb{R}^n,\, W,\, \delta,\, \mathbb{R}^n)$ is defined, where

$$\delta := \{(x_0,\, (\mu,\, \nu),\, x_1(x_0,\, \mu))| \\ x_0 \in \mathbb{R}^n,\; \mu \in U,\; \nu \in Y_1(x_0,\, \mu)\}\ . \quad (13)$$

Remark 1. The scenario fits in the framework of hybrid automata; see e. g. [al93]. Note that a hybrid automaton exhibiting the above dynamics will in general be *nonlinear*: roughly speaking, linearity of a hybrid automaton demands the continuous next state relation to be expressed by linear inequalities and affine maps. In our framework this would require $x_1(\,\cdot\,,\, \mu)$ to be affine w. r. t. the first argument. This cannot be expected: we do not even know an explicit representation of $\tau_1(\,\cdot\,,\, \mu)$, since the latter involves nonlinear equations we cannot solve analytically.

Remark 2. When discrete outputs are to be generated by the continuous output y^c crossing the boundary of certain polyhedra, an affine transformation of the continuous output space $\mathbb{R}^p$ can be used to adapt the situation to the above scenario. This also includes the case where the continuous output components are compared with threshold values.

4 Approximating switched linear systems

As δ is not known in explicit form, we cannot directly synthesize a control scheme for $P = (\mathbb{R}^n,\, W,\, \delta,\, \mathbb{R}^n)$. Thus, we suggest to construct a conservative approximation, based on partitioning the state space and sampling the continuous subsystems. To some extent, our method can be seen as an application of [pu96]. However, the situation of switched linear systems allows a large number of simplifications. These in turn enable a representation which can be more or less immediately converted into a computer program within a standard environment (e. g. "Mathematica" or "Matlab").

In order to end up with a finite procedure, we assume for all $\mu \in U$:

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

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

While (A1) is essential, (A2) is meant to facilitate implementation.

4.1 Choosing a suitable sampling rate

Sampling the differential equation at a sampling rate of $1/T \in \mathbb{R}^+$ for an input $\mu \in U$ by

$$f_{T,\mu}(x_0) := \exp(A(\mu)\, T)\, x_0 + \\ A(\mu)^{-1}\, (\exp(A(\mu)\, T) - I_n)\, B(\mu) \quad (14)$$

yields $f^r_{T,\mu}(x_0) = \varphi(r\, T,\, x_0,\, \mu)$ for all $r \in \mathbb{N}_0$. As we are looking for a conservative approximation, we also need to consider how the state evolves between the sampling instants. For a given $\rho > 0$, a sampling rate $1/T$ is choosen, such that for all $r \in \mathbb{N}_0$, $x_0 \in \mathbb{R}^n$, $\mu \in U$, $\tau,\ 0 \le \tau \le T$, the following implication holds:

$$f^r_{T,\mu}(x_0) \in R(\mu) \quad \Longrightarrow \\ ||\varphi(r\, T + \tau,\, x_0,\, \mu) - f^r_{T,\mu}(x_0)||_\infty \le \rho\ . \quad (15)$$

A suitable T can be constructed as follows: Denote the box around a bounded subset $R \subseteq \mathbb{R}^n$ with an "extra safety distance" ρ by:

$$\mathcal{S}(R,\, \rho) := \{\xi|\ \inf_{\zeta \in R} |e_i^\top\, (\zeta - \xi)| \le \rho\ \forall\, i\}\ . \quad (16)$$

Since $f(\,\cdot\,,\, \mu)$ is linear, we can compute the maximum derivative $d_{max}(\mu) = \max\{||f(\xi,\, \mu)||_\infty|\ \xi \in \mathcal{S}(R(\mu),\, \rho)\}$ by checking the vertices of $\mathcal{S}(R(\mu),\, \rho)$. Observe $d_{max}(\mu) > 0$ from (A2). Let $d_{max} := \max\{d_{max}(\mu)|\ \mu \in U\}$ and define the sampling interval by

$$T := \rho / d_{max}\ . \quad (17)$$

We now prove by contradiction that (17) guarantees (15). Therefore, assume the existence of some r, x_0, μ with $f^r_{T,\mu}(x_0) \in R(\mu)$, such that the right hand side of (15) does not hold. Hence, there exists a minimal τ, $0 < \tau < T$, such that the norm in (15) equals ρ. As $f^r_{T,\mu}(x_0) \in R(\mu)$, observe that $\varphi(r\, T + \tilde\tau,\, x_0,\, \mu) \in \mathcal{S}(R(\mu),\, \rho)$ for all $\tilde\tau \le \tau$ and therefore $||f(\varphi(r\, T + \tilde\tau,\, x_0,\, \mu),\, \mu)||_\infty \le d_{max}$ for all $\tilde\tau \le \tau$. By integrating $f(\varphi(r\, T + \tilde\tau,\, x_0,\, \mu),\, \mu)$ over $0 \le \tilde\tau \le \tau$ one obtains $||\varphi(r\, T + \tau,\, x_0,\, \mu) - f^r_{T,\mu}(x_0)||_\infty \le \tau\, d_{max} < T\, d_{max} = \rho$. This contradicts the assumption, hence (15) holds for the sampling interval defined by (17).

4.2 Tracing the state by sampling

Pick any $X_0 \subseteq \mathbb{R}^n$ and let

$$F_{T,\mu}(X_0) := f_{T,\mu}(X_0 \cap R(\mu)) \cap R(\mu)\ . \quad (18)$$

By equation (15), the following implication holds for every state trajectory $x(\,\cdot\,) = \varphi(\,\cdot\,,\, x_0,\, \mu)$ with initial state $x_0 \in X_0$ and every $t,\ r$ such that $r\, T \le t \le (r+1)\, T$:

$$x(\tau) \in R(\mu)\ \forall\, \tau \le t \\ \Longrightarrow \quad x(t) \in \mathcal{S}(F^r_{T,\mu}(X_0),\ \rho)\ . \quad (19)$$

In other words: starting at any initial state in X_0, as long as the state evolves within $R(\mu)$ and therefore may generate an output event, we can conservatively estimate the state by the boxes from (19). Vice versa, when $\mathcal{S}(F^r_{T,\mu}(X_0),\ \rho)$ happens to be empty for some r, we conclude that no output events are generated for any $t \ge r\, T$. Note that, if X_0 is a polyhedron, computing the boxes $\mathcal{S}(F^r_{T,\mu}(X_0),\ \rho)$ is straightforward.

4.3 Partitioning the state space

The continuous state space $\mathbb{R}^n$ is partitioned by a disjoint union of polyhedra q_j, $j \in \mathbb{N}$, each of them bounded: $\mathbb{R}^n = \cup_{j\in\mathbb{N}} q_j$, $q_j \cap q_i = \emptyset$ for all $i \neq j$. As an example, consider an equidistant rectangular grid. For all $\mu \in U$, assume $J(\mu) := \{j |\, q_j \cap R(\mu) \neq \emptyset\}$ to be finite. Let $J' := \cup_{\mu\in U} J(\mu)$, $q_0 := \cup_{j\notin J'} q_j$, $J := J' \cup \{0\}$. The finite set J will form the state space of the approximation: when the approximation is in state $j \in J$, this corresponds to the exact state being an element of q_j.

4.4 Approximating δ

We now ask the question: "when applying input μ to the state $x_0 \in q_{j_0}$ results in output ν, in which polyhedra might the next state x_1 be found?" To answer this, pick any $(x_0,\, (\mu,\, \nu),\, x_1) \in \delta$ and $j_0,\ j_1 \in J$ such that $x_0 \in q_{j_0}$, $x_1 \in q_{j_1}$. Assume first that $\nu \neq 0$, hence $x_0 \in R(\mu)$. The output ν determines the surface $\partial R_\nu(\mu)$ of $R(\mu)$ which the state trajectory is about to cross:

$$x_1 \in \partial R_\nu(\mu) := \{\xi |\, \xi \in R(\mu),\, e_\nu^\top\, g(\xi,\, \mu) = 0\}\,. \qquad (20)$$

Then *both* of the following conditions hold:

(C1) $\min\{e_\nu^\top\, C(\mu)\,(A(\mu)\,\xi + B(\mu))|$
$\qquad\qquad \xi \in \partial R_\nu(\mu) \cap q_{j_1}\} < 0\,,$

(C2) $\exists\, r:\ \mathcal{S}(F^r_{T,\mu}(q_{j_0}),\ \rho) \cap \partial R_\nu(\mu) \cap q_{j_1} \neq \emptyset\,.$

To prove (C1), observe that the set to be minimized consists of the derivatives of all state trajectories when crossing the surface $\partial R_\nu(\mu)$ at a state $\xi \in \partial R_\nu(\mu) \cap q_{j_1}$. If all such derivatives are positive, no trajectory starting within $R(\mu)$ can cross $\partial R_\nu(\mu) \cap q_{j_1}$, and the output ν cannot occur. Hence, the minimum is not positive. To prove the minimum to be negative, the same conclusion can be drawn, taking into account that $f(\,\cdot\,,\, \mu)$ is linear. (C2) can be inferred from (19): choose r such that $r\,T \leq \tau_1(x_0,\ \mu) < (r+1)\,T$; hence $x_1 \in \mathcal{S}(F^r_{T,\mu}(q_{j_0}),\ \rho)$.

While the implementation of a procedure to check (C1) is based on the vertices of $\partial R_\nu(\mu)$, (C2) demands more attention. By (A2), only a finite number of sampling steps are to be considered. If $x^\star_\mu \notin R(\mu)$, then there obviously exists an $r^+ \in \mathbb{N}$ such that $\mathcal{S}(F^r_{T,\mu}(q_{j_0}),\ \rho) = \emptyset$ for all $r \geq r^+$. If $x^\star_\mu \in R(\mu)$, focus on an invariant domain of attraction $G(\mu) \subseteq \mathcal{S}(\{x^\star_\mu\},\ \rho)$. Here, $G(\mu)$ is an ellipsoid which can be constructed by the Lyapunov method. Then there exists an $r^+ \in \mathbb{N}$ such that $f^r_{T,\mu}(q_{j_0}) \subseteq G(\mu)$ for all $r \geq r^+$, hence $\mathcal{S}(F^r_{T,\mu}(q_{j_0}),\, \rho) \subseteq \mathcal{S}(\{x^\star_\mu\},\, 2\,\rho)$ for all $r \geq r^+$. Therefore, (C2) implies

(C2') $[\,\cup_{r\leq r^+}\mathcal{S}(F^r_{T,\mu}(q_{j_0}),\ \rho)\ \cup \mathcal{S}(\{x^\star_\mu\},\, 2\,\rho)\,]$
$\qquad\qquad \cap\, \partial R_\nu(\mu)\ \cap\ q_{j_1} \neq \emptyset\,,$

hence, (C2) can be conservatively investigated by a procedure for checking (C2').

We now treat the case $\nu = 0$. This implies *at least one* of the following conditions to hold:

(C3) $j_0 \notin J(\mu)\,,$

(C4) $x^\star_\mu \in R(\mu)$ and $\forall\, r:\ \mathcal{S}(F^r_{T,\mu}(q_{j_0}),\ \rho) \neq \emptyset\,.$

A finite procedure for checking (C4) can be implemented in analogy to the one for (C2').

The finite state machine $P_{ca} := (J,\, W,\ \delta_{ca},\, J)$ is then defined in terms of (C1) to (C4): let $d = (j_0,\, (\mu,\, \nu),\, j_1)$ be a transition in δ_{ca} if and only if either

$\nu \neq 0$, $x^\star_\mu \notin R(\mu)$, $(C1)$ and $(C2)$,
or $\nu \neq 0$, $x^\star_\mu \in R(\mu)$, $(C1)$ and $(C2')$,
or $\nu = 0$, $j_0 = j_1$, and $(C3)$ or $(C4)$.

Let $\mathfrak{B}_{ca} \cong P_{ca}$ be the behaviour realized by P_{ca}. Then, by construction, $\mathfrak{B}_{ca} \supseteq \mathfrak{B} \cong P$, i. e. $\mathfrak{B}_{ca}$ is a conservative approximation of $\mathfrak{B}$. Obviously P_{ca} is finite.

4.5 A past induced realization

Supervisor synthesis will be based on a nonblocking past induced realization. Unfortunately, P_{ca} is *not* past induced. As we want our partition to be reasonably fine, we expect a "large" state space J. In examples for $n = 3$, a rectangular grid with 10 relevant gridpoints per axis was used. This resulted in $|J| = 10^3$; computing δ_{ca} caused no performance related problems. On the other hand, turning a state machine of this size into a past induced realization is in general not feasible. For this reason, we apply another conservative approximation scheme, namely "l-complete approximations", where $l \in \mathbb{N}$ is seen as a design parameter. Formally, l-complete approximations convert any state machine P with finite external signal space into a past induced nonblocking state machine P_l with finite state space. Denoting the induced external behaviours by $\mathfrak{B}$ and $\mathfrak{B}_l$ respectively, the crucial property of l-complete approximations is that $\mathfrak{B} \subseteq \mathfrak{B}_{l'} \subseteq \mathfrak{B}_l$ holds for all $l' \geq l$. Hence, we expect the accurary of the approximation to increase when the parameter l is increased. See [mo98a] for a discussion of l-complete approximations within the behavioural framework, addressing a general class of hybrid systems. Since the conservative approximation P_{ca} is already finite, the computation of a state machine $P_{ca,l} = (Z_l,\, W,\ \delta_{ca,l},\ Z_0)$ realizing the l-complete approximation $\mathfrak{B}_{ca,l}$ of $\mathfrak{B}_{ca}$ is straightforward (see [mo98a], Proposition 2, Theorem 2). It has to be mentioned, that increasing l not only increases the accuracy, but also the size of the realization state space Z_l. Here, the worst case is $|Z_l| = |W|^l$. However, since $|J|$ is expected to be "large", this is still better than the worst case $2^{|J|}$ when trying to construct an exact past induced realization.

5 Supervisory control

Roughly speaking, a supervisor's task is to prevent the switched linear system modelled by the state machine P from evolving on trajectories which are deemed to be unacceptable – the supervisor is meant to suitably restrict the behaviour $\mathfrak{B} \cong P$. We first synthesize a supervisor for the approximation $P_{ca,l}$ of P by employing a modified version of *Ramadge*'s and *Wonham*'s theory. Then, we observe that the supervisor obtained for $P_{ca,l}$ does indeed solve the problem for P. As a

similar solution procedure can be found in [rai98] and [mo98a], this section is only intended to give rough overview.

Denote the acceptable behaviour by $\mathfrak{B}_{spec}$ and assume it to be realized by a nonblocking past induced finite state machine $P_{spec} = (X_{spec}, W, \delta_{spec}, X_{spec_0})$. First, we remove all unacceptable trajectories by intersecting $\mathfrak{B}_{ca,l}$ and $\mathfrak{B}_{spec}$. It is a well known fact that the parallel composition of two realizations realizes the intersection of their behaviours, i. e. $B_{\|} := \mathfrak{B}_{ca,l} \cap \mathfrak{B}_{spec} \cong P_{\|} := (Q, W, \lambda, Q_0) := P_{ca,l} \parallel P_{spec}$. Clearly, $\mathfrak{B}_{\|}$ meets the specifications, but forming the parallel composition does not take into account the input/output structure of I/S/- machines. Thus, we need to refine the mechanism of interaction. While the trajectory evolves, the supervisor is only allowed to "disable" input signals in an explicit manner. In turn, this may prevent certain output signals from occuring, but the latter cannot be disabled individually: when preventing the signal $(\mu, \nu) \in U \times Y$, this can only be done by preventing all external signals $W_\mu := \{(\mu, \tilde{\nu}) | \ \tilde{\nu} \in Y\}$ simultaneously. Definition 5 formalizes the desired mechanism of interaction.

Definition 5. Let $d_1 = (z_1, w_1, z_1') \in \delta_{ca,l}$ and $d_2 = (z_2, w_2, z_2') \in \delta_{ca,l}$. The transitions d_1 and d_2 are called *partners*, if $z_1 = z_2$ and $w_1, w_2 \in W_\mu$ for some $\mu \in U$. $\tilde{P} = (\tilde{Q}, W, \tilde{\lambda}, \tilde{Q_0})$ is called a *substructure* of $P_{\|}$ w. r. t. $P_{ca,l}$, if $\tilde{\lambda} \subseteq \lambda$, $\tilde{Q}_0 \subseteq Q_0$, and a transition $((z, x_{spec}), w, (z', x'_{spec})) \in \lambda$ can only be an element in $\tilde{\lambda}$, if for every partner (z, w', z'') of (z, w, z') there exists a transition $((z, x_{spec}), w', (z'', x''_{spec}))$ in $\tilde{\lambda}$. □

The results of [mo98a] can be summarized as follows: if there is a nonblocking substructure of $P_{\|}$, then the least restrictive nonblocking substructure, denoted by P_{sup}, exists uniquely. Assuming existence, P_{sup} can be synthesized by a fixed-point algorithm. This procedure has been coded in C++ with an object oriented architecture [oy98]. If no nonblocking substructure of $P_{\|}$ exists, the supervisory control problem has no solution for $P_{ca,l}$. This implies that either the approximation $P_{ca,l}$ 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. This can be done by choosing a finer state partition or by increasing l.

Let $\mathfrak{B}_{sup}$ denote the behaviour realized by P_{sup}. Obviously, $\mathfrak{B}_{sup}$ is a subset of $\mathfrak{B}_{spec}$. Hence, when interconnecting P and P_{sup} by parallel composition, the closed loop behaviour $\mathfrak{B}_{cl} = \mathfrak{B} \cap \mathfrak{B}_{sup}$ is a subset of $\mathfrak{B}_{spec}$. Furthermore, it can be seen that $P \parallel P_{sup}$ is nonblocking and that $P \parallel P_{sup}$ itself is a substructure of $P \parallel P_{sup}$ w. r. t. P. As $P_{ca,l}$ and P_{spec} are past induced, so is $P_{\|}$ and all its substructures, hence P_{sup} is past induced too. From an application point of view, this is exactly what we are looking for: (i) the specifications are met; (ii) we do not run into a deadlock situation; (iii) when the input μ is disabled, all signals W_μ are disabled simultaneously; (iv) we can trace the supervisors state without looking at the hidden interna (e. g. continuous state variables) of P.

6 Example

We consider a thermal switched-server system consisting of three plates and a radiator, as described in [fr98]. 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 keep the temperatures of all plates in a specified range. In [fr98] a rule based switching strategy is developed by intuition. A formal verification can be found in [mo98b]. However, we feel that it is a good idea to illustrate our supervisor synthesis method by an example where some knowledge about the expected closed loop behaviour is available.

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 $\alpha_r, \ \alpha_e \in \mathbb{R}^+$; the specified range of allowed temperatures $[\beta_-, \beta_+] \subset \mathbb{R}$; it is assumed that the initial temperatures lie within $(\beta_0, \beta_+) \subset \mathbb{R}$. The temperature $x_i(\,\cdot\,)$ of plate $i \in \{1, 2, 3\}$ is modelled either by equation (21) when it is heated or by equation (22) when it is not heated:

$$\dot{x}_i(t) = \alpha_r \ (\beta_r - x_i(t)) + \alpha_e \ (\beta_e - x_i(t)) \ , \qquad (21)$$

$$\dot{x}_i(t) = 2 \ \alpha_e \ (\beta_e - x_i(t)) \ . \qquad (22)$$

Observe $x_i(t) \equiv \beta_m := (\alpha_r \ \beta_r + \alpha_e \ \beta_e)/(\alpha_r + \alpha_e)$ to be a stable equilibrium for a heated plate, and $x_i(t) \equiv \beta_e$ for one which is not heated. We assume parameter values such that $\beta_e < \beta_- < \beta_0 < \beta_+ < \beta_m < \beta_r$ holds.

Whenever the temperature of plate $\nu_{idx} \in \{1, 2, 3\}$ equals the thresholds $\nu_{val} \in \{\beta_0, \beta_+\}$, the output signal $(\nu_{idx}, \nu_{val}) \in Y' = \{1, 2, 3\} \times \{\beta_0, \beta_+\}$ is generated. In response, the supervisor may disable certain discrete input signals from $\mu \in U' = \{1, 2, 3, \ 4\}$, where $\mu = 4$ is interpreted as "radiator off", while $\mu < 4$ is interpreted as "radiator positioned at plate μ". If more than one input signal is enabled, selection is instantaneous — either at random or by some higher level control device. We look for a supervisor such that the closed loop system exhibits the following properties:

(i) All temperatures are to be kept within $[\beta_-, \ \beta_+]$.

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

(iii) No reheating process must be started for a plate at a temperature above β_0.

High frequency chattering phenomena are avoided, since by (ii) and (iii) the duration of any reheating process has a positive lower limit. When converting the scenario into a switched linear system as defined in section 3, we observe: 8 relevant boxes as output signal generating polyhedra $R(\,\cdot\,)$; 7 outputs, but less than 5 possible at each state; 32 inputs, but only 4 of them not resulting immediately in an error output. We choose a rectangular state partition, involving $|J| = 10^3$ states in P_{ca}. The specification can be realized using the same state space as

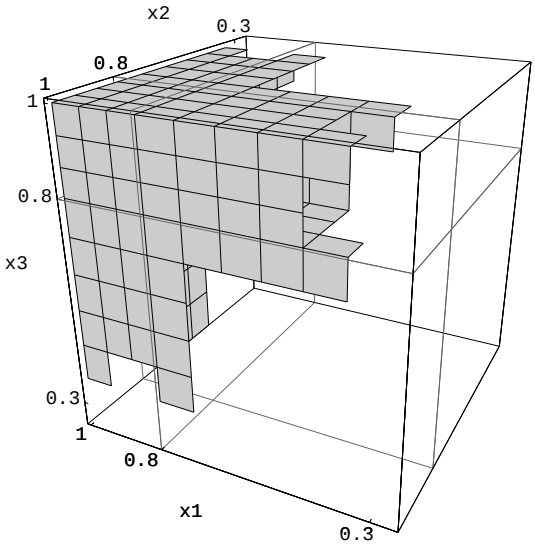

Figure 1: Closed loop reachable states

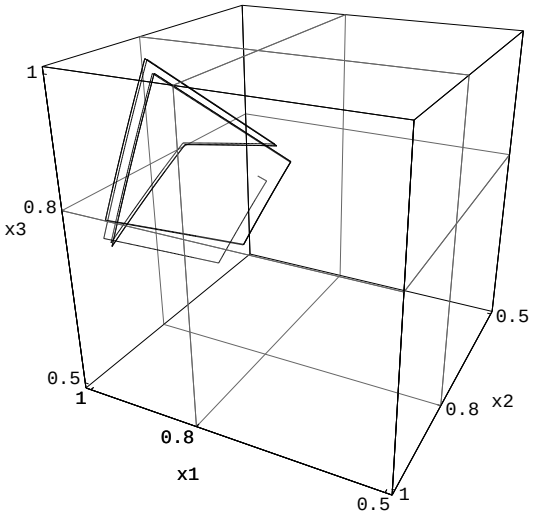

Figure 2: Closed loop simulation

$P_{ca,l}$ when $l \geq 2$, avoiding the quite expensive parallel composition $P_{ca,l} \parallel P_{spec}$. At $l = 5$, the state space of $P_{ca,l}$ counts 130×10^3 states, and supervisor synthesis succeeds for the following parameter values: $\beta_e = 0.1$, $\beta_r = 2.5$, $\alpha_e = 0.2$, $\alpha_r = 1.0$, $\beta_+ = 1.0$, $\beta_0 = 0.8$ and $\beta_- = 0.3$. Figure 1 is a conservative approximation of reachable states of $P \parallel P_{sup}$ based on $P_{ca,l} \parallel P_{sup}$. Figure 2 shows a sample trajectory of the closed loop system.

7 Conclusions

In this contribution, we suggest an approach for synthesizing supervisory control for switched linear systems based on two conservative approximation techniques. First, the switched linear system is approximated by sampling and state space partitioning. This results in a finite state machine which, in general, is not past induced. Second, applying l-completion as another conservative approximation technique converts the former result in a past induced finite state machine. This enables 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 involved algorithms can be carried out by finite procedures.

References

[al93] Alur, R., Courcoubetis, C., Henzinger, T. A., Ho, P.-H.: "Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems", in Grossman R. L., Nerode, A., Ravn, A. P., Rischel, H. (Eds.): *Hybrid systems*, LNCS 736, pp. 209-229, Springer, Berlin, (1993).

[fr98] Franke, D., Moor, T.: "Combined rule- and model-based design of a hybrid thermal process", *Proc. CESA98*, pp. 630-634, Nabeul-Hammamet, Tunesien, (1998).

[mo98a] Moor, T., Raisch, J., O'Young, S. D.: "Supervisory control of hybrid systems via l-complete approximations", *Proc. WODES98*, pp. 426-431, (1998).

[mo98b] Moor, T., Raisch, J.: "Estimating reachable states of hybrid systems via l-complete approximations", *Proc. SSCC98*, **Vol. 3**, pp. 30-34, Durban, Southafrica, (1998).

[oy98] O'Young, S. D.: "Hybrid RTSS", *Internal Report, Faculty of Engineering, Memorial University of Newfoundland*, (1998).

[pu96] Puri, A., Borkar, V., Varaiya, P.: "ϵ-Approximation of differential inclusions", in Alur, R., Henzinger, T. A., Sontag, E. D. (Eds.): *Hybrid systems III*, LNCS 1066, pp. 362-376, Springer, Berlin, (1996).

[rai97] Raisch, J., O'Young, S. D.: "A totally ordered set of discrete abstractions for a given hybrid or continuous system", in Antsaklis, P., Kohn, W., Nerode, A., Sastry, S. (Eds.): *Hybrid Systems IV*, LNCS 1273, pp. 342-360, (1997).

[rai98] Raisch, J., O'Young, S. D.: "Discrete approximation and supervisory control of continuous systems", *IEEE Transactions on Automatic Control, Special issue on hybrid systems*, **Vol. 43**, (1998).

[ram89] Ramadge, P. J., Wonham, W. M.: "The control of discrete event systems", *Proceedings of the IEEE*, **Vol. 77**, pp. 81-98, (1989).

[tit94] Tittus, M., Egardt, B.: "Control-law synthesis for linear hybrid systems", *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 961-966, (1994).

[wi89] Willems, J. C.: "Models for dynamics", *Dynamics Reported*, **Vol. 2**, pp. 172-269, (1989).

[wi91] Willems, J. C.: "Paradigms and puzzles in the theory of dynamic systems", *IEEE Transactions on Automatic Control*, **Vol. 36**, No. 3, pp. 258-294, (1991).