Approximation of multiple switched flow systems for the purpose of control synthesis

Thomas Moor [1] Jörg Raisch [2]

Abstract

Switched flow systems exhibit a continuous state evolving w.r.t. continuous time. However, a purely discrete interface to the environment is provided, acting on input events by switching between a finite number of flows and generating output events whenever the continuous state crosses certain boundaries. Here, a practically important task is the synthesis of a supervisory controller enforcing a specification in terms of input and output events to be met. A well known approach is so called approximation based synthesis, where the switched flow system is approximately realized by a finite automaton. In the situation of multiple switched flow systems the question arises first, how to approximate the individual systems, and second, how to compose the approximations in order to finally apply a synthesis scheme. It turns out that retaining clock time on the approximation level leads to a suitable overall procedure.

Keywords: hybrid systems, discrete event systems, discrete approximations, supervisory control synthesis.

1 Introduction

The switched flow systems under consideration exhibit a continuous state $x(t) \in \mathbb{R}^n$ evolving w.r.t. continuous time $t \in \mathbb{R}_0^+$. The continuous dynamics are given by a finite number of flows. Discrete input events implement a switching between the flows such that there is just one flow enabled at every time $t \in \mathbb{R}_0^+$. Discrete output events are generated whenever the state crosses certain boundaries, e.g. by triggering threshold values. The state is treated as an internal variable, not "visible" from the environment point of view, while input and output events are external, i.e. are respectively applied by and reported to the environment. Involving both discrete and continuous variables, switched flow systems belong to the class of so called hybrid systems.

In the situation of supervisory control the environment will consist of a controller, which for the scope of this paper is restricted to be a discrete event system (DES) realized by a finite automaton. Then, the problem of supervisory control synthesis is about the construction of a controller that enforces a switched flow system to meet a given closed loop specification. However, rather restrictive conditions apply to hybrid systems on which synthesis can be treated directly; e.g. [15]. Therefore it is suggested to approximate the switched flow system by some finite automaton and then to apply known synthesis procedures from DES theory; e.g. [5, 6, 9, 11, 12, 13].

In [9, 11], we propose an approximation based approach for supervisory control synthesis for a general class of hybrid systems within the framework provided by Willems' behavioural systems theory (e.g. [16]), and Ramadge and Wonham's supervisory control theory (e.g. [14]). While [9, 11] address the control theoretic aspects, in [10] computational issues are discussed, assuming that the flows involved are induced by linear time invariant differential equations. This paper addresses *multiple switched flow systems* in the context of supervisory controller synthesis. By a multiple switched flow system we refer to a system made up by a number of individual *sub-systems*, each of them a switched flow system. Of course, a multiple switched flow system could be handled by first of all combining the sub-systems to a single switched flow system. However, this is not recommended because the approximation procedure is of high order complexity w.r.t. the dimension of the state space. Clearly, the sensible aim is to first approximate each individual sub-system in order to then compose the approximations. While each sub-system evolves independently each of the involved state variables share only the one real time axis $\mathbb{R}_0^+$. Hence, when applying an approximation scheme to individual sub-systems, it should retain the timing information to some extent. In the framework of DES this can be done by referring to clock time instead of logic time. By doing so, approximating individual sub-systems becomes a suitable foundation for the synthesis of a supervisory controller running all sub-systems simultaneously.

This paper is organized as follows. In Section 2, we give a definition of switched flow systems. Using the notion of flow pipes as introduced in [4], Section 3 restates the approximation scheme presented in [10] from a more general point of view. In Section 4, the flow pipe estimate will be represented by a finite automaton. The situation of multiple switched flow systems is discussed in Section 5; we show how to compose separate approximations of individual subsystems. It is then seen that the overall approximation procedure is conservative w.r.t. the external behaviour and thus meets the requirements of the synthesis procedure provided in [9, 11].

[1] Research School of Information Sciences and Engineering, Australian National University, Canberra, email: thomas.moor@anu.edu.au.
Research partially supported by US Office of Naval Research, Grant N 00014-98-1-0535

[2] Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg, FR Germany, email: raisch@mpi-magdeburg.mpg.de

2 Switched flow systems

In this section, a system class is introduced in which the external signals are discrete, while the internal dynamics are represented by a finite number of continuous flows. The suggested scenario is closely related to the framework of hybrid automata; e.g. [1]. But due to our focus on controller synthesis, we require an explicit notion of inputs and outputs.

Let $\Phi(\mu, \cdot, \cdot)\colon \mathbb{R}^n \times \mathbb{R}_0^+ \to \mathbb{R}^n$ be a semi-flow for every $\mu \in U$, $|U| \in \mathbb{N}$; i.e. $\Phi(\mu, \cdot, \cdot)$ is continuous w.r.t. both arguments separately and $\Phi(\mu, \xi, 0) = \xi$, $\Phi(\mu, \xi, t_1 + t_2) = \Phi(\mu, \Phi(\mu, \xi, t_1), t_2)$ for all $\xi \in \mathbb{R}^n$, $t_1,\ t_2 \in \mathbb{R}_0^+$. The parameter μ implements a switching between the semi-flows. The parameter set U is therefore seen as the set of input events that the switched flow system will accept. If at time $t_0 \in \mathbb{R}_0^+$ the system is in state $x(t_0) \in \mathbb{R}^n$ and the input event $\mu \in U$ is applied, the state $x(t)$, $t \geq t_0$, will evolve according to $\Phi(\mu, x(t_0), t - t_0)$ until the next input event takes place.

For each $\mu \in U$ let $\mathit{Inv}(\mu) \subseteq \mathbb{R}^n$ denote a subset of the state space. Applying the input μ will be only allowed at time t if the system is in some state $x(t) \in \mathit{Inv}(\mu)$. Moreover, if $\mu \in U$ is the most recently applied input event, at the time the trajectory is about to leave $\mathit{Inv}(\mu)$ an output event $\nu \in Y$, $|Y| \in \mathbb{N}$, is generated. This indicates that a new input event is required. It will be the supervisory controller's task to apply suitable input events such that in the closed loop situation the state evolves within $\mathit{Inv}(\mu)$ as long as $\mu \in U$ is the most recent input event. Thus, $\mathit{Inv}(\mu)$ is referred to as the invariant associated with the input event μ.

To formalize the generation of output events, focus is on the situation where at time $t_0 \in \mathbb{R}_0^+$ the state is $x(t_0) = \xi_0 \in \mathbb{R}^n$ and the input $\mu \in U$ has been applied. Provided that no more input events take place, the time at which an output event occurs is given by

$$t'(\xi_0, \mu) := \sup\{t |\ \Phi(\mu, \xi_0, \tau) \in \mathit{Inv}(\mu)\ \forall\, 0 \leq \tau < t\}\ . \quad (1)$$

Here, $t'(\xi_0, \mu) = \infty$ indicates that the state trajectory evolves within $\mathit{Inv}(\mu)$ for all future and that no output event will be generated. Assuming $\mathit{Inv}(\mu)$ to be an open set, $t'(\xi_0, \mu) = 0$ corresponds to $\xi_0 \notin \mathit{Inv}(\mu)$. In the case $0 < t'(\xi_0, \mu) < \infty$, an output event will occur when the system is in state

$$x'(\xi_0, \mu) := \Phi(\mu, \xi_0, t'(\xi_0, \mu))\,. \quad (2)$$

If generated at all, an output event represents a quantized version of the current state: given a finite cover $\cup_{\nu \in Y} \mathit{Grd}(\mu, \nu)$ of the boundary $\partial \mathit{Inv}(\mu)$, at time $t'(\xi_0, \mu)$ an output event $\nu \in y'(\xi_0, \mu)$ will occur, whereby

$$y'(\xi_0, \mu) := \{\nu |\ x'(\mu, \xi_0) \in \mathit{Grd}(\mu, \nu)\} \subseteq Y\,. \quad (3)$$

Given Φ, Inv and Grd, the above describes the relation between input events, output events and the evolution of the continuous state. The triple $(\Phi, \mathit{Inv}, \mathit{Grd})$ is referred to as a *switched flow system*. A trajectory of $(\Phi, \mathit{Inv}, \mathit{Grd})$ is defined to be a triple (u, y, x) consisting of an input, an output, and a state component, all of them with evolving w.r.t. the continuous time axis $\mathbb{R}_0^+$. As discrete events only occur at discrete points on the time axis, the sets U and Y are extended by an additional symbol $\mathit{Na} \notin U \cup Y$ where '$u(t) = \mathit{Na}$' reads 'no input event at time t'. Let $U^{\#} = U \cup \{\mathit{Na}\}$ and $Y^{\#} = Y \cup \{\mathit{Na}\}$. Then, any trajectory $(u, y, x)\colon \mathbb{R}_0^+ \to U^{\#} \times Y^{\#} \times \mathbb{R}^n$ of $(\Phi, \mathit{Inv}, \mathit{Grd})$ is required to fulfill all of the following conditions w.r.t. some $(t_k)_{k \in K}$, $K \subseteq \mathbb{N}_0$, $T := \{t_k |\ k \in K\} \subset \mathbb{R}_0^+$, $t_{k+1} \geq t_k$, $t_0 = 0$ and all $k \in K$, $t \in \mathbb{R}_0^+$:

(A1) $y(t) \neq \mathit{Na} \Rightarrow u(t) \neq \mathit{Na} \Leftrightarrow t \in T; \quad y(t_0) = \mathit{Na}$

(A2) $t \in [t_k, t_{k+1}] \Rightarrow$
$x(t) = \Phi(u(t_k), x(t_k), t - t_k) \in \overline{\mathit{Inv}}(u(t_k))$

(A3) $|K| \in \mathbb{N}_0$ and $t \in [t_{\max(K)}, \infty) \Rightarrow$
$x(t) = \Phi(u(t_k), x(t_k), t - t_k) \in \overline{\mathit{Inv}}(u(t_k))$

(A4) $y(t_{k+1}) \neq \mathit{Na} \Rightarrow t_{k+1} = t'(x(t_k), u(t_k))$

(A5) $t_{k+1} = t'(x(t_k), u(t_k)) \Rightarrow$
$y(t_{k+1}) \in y'(x(t_k), u(t_k))$

The set of all trajectories which fulfill the above is referred to as the *full behaviour* $\mathfrak{B}(\Phi, \mathit{Inv}, \mathit{Grd})$ of the switched flow system $(\Phi, \mathit{Inv}, \mathit{Grd})$. The set of all pairs of input and output signals $(u, y)\colon \mathbb{R}_0^+ \to U^{\#} \times Y^{\#}$ such that $(u, y, x) \in \mathfrak{B}(\Phi, \mathit{Inv}, \mathit{Grd})$ holds true for some x is referred to as the *external behaviour* $\mathfrak{B}_{ex}(\Phi, \mathit{Inv}, \mathit{Grd})$. While the state x is considered to evolve "inside" the switched flow system, the external signals u and y are "visible" to the environment and thus to any supervisory controller interconnected with the switched flow system. Hence, the external behaviour $\mathfrak{B}_{ex}(\Phi, \mathit{Inv}, \mathit{Grd})$ plays an important role in supervisory control synthesis and Willems' "behavioural systems theory" forms a natural framework for our investigations [9, 11, 16]. However, for the scope of this paper, we stick to the realization level: all behaviours we deal with will be realized either by switched flow systems or finite automata.

When the flows are induced by linear time invariant differential equations $(\Phi, \mathit{Inv}, \mathit{Grd})$ is referred to as switched linear system. For the scope of this paper, the sets $\mathit{Inv}(\mu)$ and $\mathit{Grd}(\mu, \nu)$ are assumed to be polyhedra whenever switched linear systems are considered. Note that the proposed view of linearity is less restrictive than the one usually applied in the framework of hybrid automata: roughly speaking, linearity of a hybrid automaton is seen w.r.t. logic time and will, in the context of switched flow systems, require $x'(\cdot, \mu)$ to be linear. However, the latter condition is by no means implied by linearity of $\Phi(\mu, \cdot, t)$. As far as continuous dynamics are concerned, switched linear systems form a much richer class than linear hybrid automata. For the scope of this paper, linearity refers to switched linear systems rather than linear hybrid automata.

3 Conservative estimates on flow pipes

Given an initial state and an input signal, the trajectory on which the switched flow system will evolve cannot in general be computed exactly, even when restricted to the linear case. Thus reliable approximation techniques are required. For both verification and controller synthesis it is highly desirable to find an estimate which is conservative in the sense that all actual trajectories of the flow will be included. Conservative estimates for certain classes of flows w.r.t. a given interval of time are provided by [4, 8]. However, when the switching is to be done by some supervisory controller which is driven by output events, an estimate of a flow w.r.t. a given invariant set rather than w.r.t. an interval of time seems to be more appropriate; e.g. [10]. In the following, we use a notion of flow pipes provided by [4], in order to restate the approximation method suggested in [10] from a more general viewpoint.

Let the input $\mu \in U$ be applied at time 0 to the switched flow system $(\Phi,\, \mathit{Inv},\, \mathit{Grd})$ and let us focus attention on some set of initial states $X_0 \subseteq \mathbb{R}^n$. Then, the *flow pipe* $\mathcal{R}(X_0,\, [0,t]) \subseteq \mathbb{R}^n$ w.r.t. the time interval $[0,t] \subset \mathbb{R}_0^+$ is defined to be the set of states reachable from some state $\xi_0 \in X_0$ by some time $\tau \in [0,t]$. Recall from [4] the following definition:

$$\mathcal{R}(X_0,\, t) := \{\Phi(\mu,\, \xi_0,\, t) |\, \xi_0 \in X_0\}\,, \tag{4}$$

$$\mathcal{R}(X_0,\, [t_0,t_1]) := \bigcup_{t\in[t_0,t_1]} \mathcal{R}(X_0,\, t)\,. \tag{5}$$

The following properties are observed immediately:

$$\mathcal{R}(X_0 \cup \tilde{X}_0,\, [t_0,t_1]) = \mathcal{R}(X_0,\, [t_0,t_1]) \cup \mathcal{R}(\tilde{X}_0,\, [t_0,t_1])\,, \tag{6}$$

$$\mathcal{R}(X_0,\, [t_0,t_1] \cup [\tilde{t}_0,\tilde{t}_1]) = \mathcal{R}(X_0,\, [t_0,t_1]) \cup \mathcal{R}(X_0,\, [\tilde{t}_0,\tilde{t}_1])\,, \tag{7}$$

$$\mathcal{R}(X_0,\, [t_0+\tilde{t}, t_1+\tilde{t}]) = \mathcal{R}(\mathcal{R}(X_0,[t_0,t_1]),\, \tilde{t}) = \mathcal{R}(\mathcal{R}(X_0,\tilde{t}),\, [t_0,t_1])\,. \tag{8}$$

This suggests the construction of an estimate of $R(X_0,\, [0,t])$ from smaller pieces. Here, it is assumed that the flow under consideration can be reliably simulated and that the reachable states can be conservatively estimated w.r.t. a “small” time interval $[0,\Delta]$ —provided that the sets of states involved belong to a certain class $\mathfrak{X} \subseteq \{X |\, X \subseteq \mathbb{R}^n\}$. More formally, it is assumed that a procedure $\hat{\mathcal{R}}$ can be established such that sets

$$\mathcal{R}(X_0,\, t) \subseteq \hat{\mathcal{R}}(X_0,\, t) \in \mathfrak{X}\,, \tag{9}$$

$$\mathcal{R}(X_0,\, [0,\Delta]) \subseteq \hat{\mathcal{R}}(X_0,\, [0,\Delta]) \in \mathfrak{X}\,, \tag{10}$$

can be computed for arbitrary $X_0 \in \mathfrak{X}$, $t \in \mathbb{R}_0^+$. Several classes $\mathfrak{X}$ are considered in the literature: polyhedra in [4, 10], griddy polyhedra in [7] and ellipsoids in [8]. Given $X_0 \in \mathfrak{X}$, $r^+ \in \mathbb{N}$, $t \in [r^+\Delta, (r^+ + 1)\Delta]$, the flow pipe w.r.t. $[0,t]$ can be conservatively estimated in terms of $\hat{\mathcal{R}}$:

$$\mathcal{R}(X_0,\, [0,t]) \subseteq \bigcup_{r=0}^{r^+} \hat{\mathcal{R}}(\hat{\mathcal{R}}(X_0,[0,\Delta]),\, r\Delta) \tag{11}$$

or

$$\mathcal{R}(X_0,\, [0,t]) \subseteq \bigcup_{r=0}^{r^+} \hat{\mathcal{R}}(\hat{\mathcal{R}}(X_0,r\Delta),\, [0,\Delta])\,, \tag{12}$$

whichever is expected to be the more accurate or performed with less computational effort. In the situation of switched linear systems, $\mathcal{R}(\,\cdot\,, r\Delta)$ is a linear map that can be applied to compact polyhedra by means of the vertices. Because [4] provides a sophisticated method for the computation of $\hat{\mathcal{R}}(X_0,\, [0,\Delta])$, it is consequently suggested in [4] to use (11) rather than (12).

As noted above, in the context with switched flow systems, the flow pipe is of interest not w.r.t. some interval on the time axis but w.r.t. the invariant $\mathit{Inv}(\mu) \subseteq \mathbb{R}^n$. Given a flow invariant domain $G(\mu)$ (i.e. $\Phi(\mu,\, \xi,\, t) \in G(\mu)$ for all $t > 0, \xi \in G(\mu)$), we suggest the following iteration:

- (B1) Assign $r := 0$.
- (B2) Assign $X_{r+1} := \hat{\mathcal{R}}(X_r,\, \Delta) \cap \overline{\mathit{Inv}}(\mu)$
- (B3) If $X_{r+1} = \emptyset$, then
  assign $r^+ := r + 1$ and terminate.
- (B4) If $X_{r+1} \subseteq G(\mu)$, then
  assign $X_{r+1} := G(\mu)$,
  assign $r^+ := r + 1$ and terminate.
- (B5) Assign $r := r + 1$ and proceed with (B2).

We comment on the above steps. At (B2) the intersection with the invariant discards any evolution of the flow outside $\overline{\mathit{Inv}}$. If it happens that either $\overline{\mathit{Inv}} \notin \mathfrak{X}$ or $\mathfrak{X}$ is not closed under intersection, a conservative estimate of $\hat{\mathcal{R}}(X_r,\, \Delta) \cap \overline{\mathit{Inv}}(\mu)$ within $\mathfrak{X}$ needs to established. E.g. this is the case when $\mathfrak{X}$ is a set of ellipsoids. However, by choosing polyhedra or griddy polyhedra, the intersection will lie within $\mathfrak{X}$ as required. At (B3) the iteration is terminated if all trajectories starting within X_0 are known to have left $\mathit{Inv}(\mu)$ at some $t \leq (r + 1)\Delta$. At (B4) the iteration is terminated if all considered trajectories have either left $\mathit{Inv}(\mu)$ or have approached $G(\mu)$ at some $t \leq (r + 1)\Delta$. In the latter case, $G(\mu)$ serves as a conservative estimate of all future evolution. If $G(\mu) \notin \mathfrak{X}$, a conservative estimate $\hat{G}(\mu) \in \mathfrak{X}$, $\hat{G}(\mu) \supseteq G(\mu)$, is to be assigned to X_{r+1} at (B4) instead.

In order to result in a finite procedure, trajectories are required either to leave $\mathit{Inv}(\mu)$ or to approach $G(\mu)$ within an uniformly bounded time t^+. While t^+ is not required explicitly, a suitable $G(\mu)$ needs to be established. In the case of an asymptotically stable linear system, ellipsoidal invariant domains of attraction $G(\mu)$ can be constructed within any neighbourhood of the unique equilibrium by solving a Lyapunov equation. If, in addition, $\mathit{Inv}(\mu)$ is bounded, a finite bound t^+ exists. Choosing $\mathfrak{X}$ to be the set of compact polyhedra, (B2) can be carried out exactly. Thus, if $G(\mu)$ is entirely outside $\overline{Inv}(\mu)$, the iteration will indeed terminate due to $(B3)$. If $G(\mu)$ is not entirely outside $\overline{Inv}(\mu)$, then still all trajectories will uniformly approach $G(\mu)$ within some time t^+, causing the iteration to terminate due to step (B4). As either the if-clauses of (B3) or (B4) become true within a finite number of steps, in the situation of switched linear systems the iteration is guaranteed to terminate faithfully.

In order to also cover the evolution of trajectories between the sampling instances $r\Delta$, the estimate $\hat{\mathcal{R}}(\cdot\ [0,\Delta])$ is applied to the finite sequence $X_0,\, X_1,\, \ldots\, X_{r+}$. The overall result of the iteration will be referred to by

$$\tilde{\mathcal{R}}(\mu,\, X_0,\, r) := \hat{\mathcal{R}}(X_r,\, [0,\Delta]) \quad \forall\, 0 \le r < r^+\,, \qquad (13)$$

$$\tilde{r}^+(\mu,\, X_0) := r^+\,. \qquad (14)$$

By construction, the union over all $\tilde{\mathcal{R}}(\mu,\, X_0,\, r)$ conservatively estimates the state as long as it evolves within $\overline{Inv}(\mu)$ and no inputs other than μ are applied.

As an example, we provide a simple version of the estimate $\hat{\mathcal{R}}(X_0,\, [0,\Delta])$. Denote the box around a bounded subset $Q \subset \mathbb{R}^n$ with an "extra safety distance" $\rho \in \mathbb{R}^+$ by:

$$\mathcal{S}(Q,\, \rho) := \{\xi |\ \inf_{\xi' \in Q} |e_i^\top (\xi' - \xi)| \le \rho\ \forall\, i\}\,. \qquad (15)$$

We assume the flow $\Phi(\mu,\ \cdot,\ \cdot)$ to be introduced by a differential equation $\dot{x}(t) = f(x(t))$, where the right hand side f is continuous. Then the maximum derivative

$$d_{max} = \sup\{||f(\xi)||_\infty |\ \xi \in \mathcal{S}(Inv(\mu),\, \rho)\} \qquad (16)$$

is finite. In the linear case, d_{max} is computed by checking the vertices of $\mathcal{S}(Inv(\mu),\, \rho)$. If the interval $[0,\Delta]$ is chosen such that $\Delta\, d_{max} \le \rho$ holds true, the state trajectory x by no component varies by more than ρ within time Δ. Then, $\hat{\mathcal{R}}(X_0,\, [0,\Delta]) := \mathcal{S}(X_0,\, \rho)$ gives a conservative estimate on $\mathcal{R}(X_0,\, [0,\Delta])$. While being less accurate than the method provided in [4], our version computes rather fast. Furthermore, as the estimates are rectangular boxes, powerful algorithms for intersection and union with griddy polyhedra exist; e.g. [3]. This is of special interest if —as in the context of supervisory control synthesis— an entire switched flow system is subject to estimation rather than a single flow pipe.

4 Representation by finite automata

In order to capture the dynamics of the overall switched flow system $(\Phi,\ Inv,\ Grd)$ it is suggested to apply the procedure given in Section 3 to a finite cover of all invariants. Since there are only a finite number of finite sequences involved, this leads to an conservative estimate of a switched flow system by a finite automaton.

When a single switched flow system is interconnected with a supervisory controller one may assume that input events occur only as immediate reply to output events. In this case, dynamics may faithfully be considered w.r.t. logic time $k \in \mathbb{N}_0$, where k counts pairs of input and output events. Thus, in the perspective of logic time, only the order of events is retained while their location on the continuous time axis $\mathbb{R}_0^+$ is entirely discarded. Note that logic time is the standard interpretation of time in the DES context. However, if multiple switched flow systems are to be considered, an output event generated by one of them may cause the supervisor to apply an input event to some other switched flow system. It then is crucial to retain timing information to at least some extent. We therefore suggest to set up an approximation that refers to clock time, by introduction of an additional event $Tick$ which presents the passing by of a certain amount of time; see [2] for a discussion of timed discrete event systems (TDES).

The construction of our finite automaton is begun by choosing a state set Z. In order to end up with a finite set Z, sets of initial states X_0 are restricted to be from some finite subset $\mathfrak{X}_0 = \{X_{0_j} |\ j \in J\} \subseteq \mathfrak{X},\ |J| \in \mathbb{N}$. Since the entire switched flow system is to be estimated, $\mathfrak{X}_0$ in turn is required to form a cover of all invariants $Inv(\mu),\ \mu \in U$; i.e. $\cup_{j\in J} X_{0_j} \supseteq \cup_{\mu \in U} Inv(\mu)$. A nearby choice of $\mathfrak{X}_0$ is given by rectangular boxes or balls based on a regular grid. Most of the time, the flow pipe estimate takes place within the class $\mathfrak{X}$ and hence is expected to be quite accurate. However, it will be necessary to break down sets of continuous states to elements of $\mathfrak{X}_0$ just before a new input is applied. In addition to the index j indicating the set of initial states X_{0_j}, a state $\zeta = (j,\, \mu,\, r) \in Z$ shall indicate the input μ which has been applied most recently and the step r of the iteration (B1)-(B5). If no input has been applied so far, this is indicated by $\mu = None$. Finally, Z is given by

$$R := \max\{\tilde{r}^+(\mu,\, X_{0_j}) |\ \mu \in U,\ j \in J\}\,, \qquad (17)$$

$$Z := J \times (U \cup \{None\}) \times \{0,\, 1,\, \ldots R\}\,. \qquad (18)$$

The transitions in the finite automaton we are about to construct will be labeled by the event set

$$W = U \cup\ Y \cup \{Tick\}\,. \qquad (19)$$

While $\mu \in U$ and $\nu \in Y$ refer to the respective input and output event of the switched flow system, $Tick$ is introduced to represent the passing of continuous time Δ.

A transition is a triple $(\zeta,\ \omega\ \zeta') \in Z \times W \times Z$, indicating that the event ω "drives" the state ζ to ζ'. Since we aim to represent the dynamics of $(\Phi,\ Inv,\ Grd)$, a transition $(\zeta,\ \omega\ \zeta') \in Z \times W \times Z$, $\zeta = (j,\, \mu,\, r)$, $\zeta' = (j',\, \mu',\, r')$, is only allowed to take place under one of the following circumstances

- (C1) Applying an input event to a state holding no recent input so far: $\mu = None$, $\omega \in U$, $X_{0_j} \cap Inv(\omega) \neq \emptyset$, $j' = j$, $\mu' = \omega$, $r' = 0$.
- (C2) Applying an input event to a state holding a recent input: $\mu \in U$, $\omega \in U$, $\tilde{\mathcal{R}}(\mu,\, X_{0_j},\, r) \cap \mathcal{X}_{0_{j'}} \cap Inv(\omega) \neq \emptyset$, $\mu' = \omega$, $r' = 0$.
- (C3) Evolution of time as long as $G(\mu)$ is not approached: $\mu \in U$, $\omega = Tick$, $r < \tilde{r}^+(\mu,\, X_{0_j})$, $\tilde{\mathcal{R}}(\mu,\, X_{0_j},\, r +$

$1) \neq \emptyset$, $j' = j$, $\mu' = \mu$, $r' = r + 1$.

(C4) Evolution of time within $G(\mu)$: $\mu \in U$, $\omega = \mathit{Tick}$, $r = \tilde{r}^{+}(\mu, X_{0_j})$, $\tilde{\mathcal{R}}(\mu, X_{0_j}, r) \neq \emptyset$, $j' = j$, $\mu' = \mu$, $r' = r$.

(C5) Generation of an output event: $\mu \in U$, $\omega \in Y$, $\tilde{\mathcal{R}}(\mu, X_{0_j}, r) \cap \mathit{Grd}(\mu, \omega) \cap \mathcal{X}_{0_{j'}} \neq \emptyset$, $\mu' = \mathit{None}$, $r' = 0$.

The set of all those transitions which agree with (C1)-(C5) is denoted by $\delta \subseteq Z \times W \times Z$. The triple (Z, W, δ) is referred to as the finite automaton representing the flow pipe estimate of $(\Phi, \mathit{Inv}, \mathit{Grd})$. As with $(\Phi, \mathit{Inv}, \mathit{Grd})$, there will be a behaviour associated with (Z, W, δ). Here, trajectories $(u, y, z)\colon \mathbb{R}_0^+ \to U^\# \times Y^\# \times Z$ are required to fulfill all of the conditions (D1) to (D4) below w.r.t. some $(t_k)_{k \in \mathbb{N}_0}$, $T := \{t_k |\, k \in \mathbb{N}_0\} \subset \mathbb{R}_0^+$, $t_{k+1} \geq t_k$, $t_0 = 0$ and for all $k \in \mathbb{N}_0$, $t \in \mathbb{R}_0^+$:

(D1) $z(t)$ is constant on $[t_k, t_{k+1})$.

(D2) $u(t) \neq \mathit{Na} \Rightarrow t \in T$; $\quad y(t) \neq \mathit{Na} \Rightarrow t \in T$.

(D3) $(\zeta, u(t_0), z(t_0)) \in \delta$ for some $\zeta = (j, \mathit{None}, 0) \in Z$; $\quad y(t_0) = \mathit{Na}$.

(D4) Either one of (D4a), (D4b), (D4c) must hold true:

- (D4a) $(z(t_k), \mathit{Tick}, z(t_{k+1})) \in \delta$, $t_{k+1} - t_k = \Delta$, $u(t_{k+1}) = \mathit{Na} = y(t_{k+1})$.
- (D4b) $(z(t_k), u(t_{k+1}), z(t_{k+1})) \in \delta$, $t_{k+1} - t_k \leq \Delta$, $y(t_{k+1}) = \mathit{Na}$.
- (D4c) There exists a $\zeta' \in Z$ such that: $(z(t_k), y(t_{k+1}), \zeta') \in \delta$, $(\zeta', u(t_{k+1}), z(t_{k+1})) \in \delta$, $t_{k+1} - t_k \leq \Delta$.

The full behaviour $\mathfrak{B}(Z, W, \delta)$ is defined to be the set of all trajectories (u, y, z) agreeing with (D1) to (D4). The external behaviour $\mathfrak{B}_{ex}(Z, W, \delta)$ is defined to be the set of all external trajectories $(u, y)\colon \mathbb{R}_0^+ \to U^\# \times Y^\#$ such that $(u, y, z) \in \mathfrak{B}(Z, W, \delta)$ for some z. Because the actual flows Φ are each estimated conservatively by $\tilde{\mathcal{R}}$, it can be seen that the finite automaton indeed conservatively approximates the switched flow system w.r.t the external behaviour:

$$\mathfrak{B}_{ex}(Z, W, \delta) \supseteq \mathfrak{B}_{ex}(\Phi, \mathit{Inv}, \mathit{Grd})\,. \tag{20}$$

In order to formally prove (20), for an arbitrary trajectory $(u, y, x) \in \mathfrak{B}(\Phi, \mathit{Inv}, \mathit{Grd})$ a z is established such that $(u, y, z) \in \mathfrak{B}(Z, W, \delta)$. We omit this straightforward but quite cumbersome technical detail.

5 Multiple switched flow systems

In this Section we focus attention on a scenario where an overall plant is formed by a finite number of individual switched flow systems $(\Phi_i, \mathit{Inv}_i, \mathit{Grd}_i)$, $i \in I$, $|I| \in \mathbb{N}$, evolving on individual trajectories. A supervisory controller is connected to the plant to form a closed loop system: whenever one of the switched flow systems generates an output event, the supervisor will apply input events to all of the switched flow systems. The aim of supervisory control is to force the closed loop to fulfill some specification. A typical specification demands certain output events not to be generated, as they correspond to undesired situations; e. g. the continuous part of the plant state evolving into a "forbidden" region. More sophisticated specifications also incorporate dynamics, implying that the occurrence of certain events affects what needs to be prevented in future; e.g. specifications that aim at forcing the closed loop into some cyclic behaviour.

Given a specification and a plant, the problem of supervisory control synthesis is about the construction of a supervisory controller, such that the closed loop indeed meets the specifications. In the situation of (multiple) switched flow systems —as in general with hybrid systems— rather restrictive conditions apply when demanding that a synthesis procedure deals with both continuous and discrete variables directly. On the other hand, so called approximation based synthesis methods are available for non-trivial flows [5, 6, 9, 11, 12, 13]. The rather simple idea behind the approximation based approach is to replace the continuous dynamics by some finite automaton and than solve the synthesis problem for the automaton by known methods from the field of DES theory; e.g. [2, 14]. However, it then needs to be shown that a solution found on approximation level can indeed be turned into a supervisor which runs the actual hybrid system according to the specification. The crucial condition required by all of [5, 6, 9, 11, 12, 13] is that the approximation needs to be conservative. Considering a multiple switched flow system, one possible approach would be first, to compose the multiple switched flow systems to a single one, second, to apply a conservative estimation scheme, and finally to run an approximation based synthesis procedure. However, the complexity of the conservative estimate $\hat{\mathcal{R}}$ tends to be of high order w.r.t. the dimension of the continuous state space and typically demands a giant number of expensive floating point operations to be carried out. Details depend on the methods used when implementing $\tilde{\mathcal{R}}$; e.g. intersection of polyhedra is of exponential order. Therefore it is suggested first to establish the conservative estimate of every individual sub-system, second, to compose the resulting finite automata and finally to apply the synthesis procedure.

In fact, it is easily observed that the latter suggestion leads to a conservative approximation of the multiple switched flow system w.r.t. the external behaviour. As the individual state trajectories evolve independently of each other, the full behaviour of the plant is

$$\mathfrak{B} := \{((u_i)_{i\in I}, (y_i)_{i\in I}, (x_i)_{i\in I})|\; (u_i, y_i, x_i) \in \mathfrak{B}(\Phi_i, \mathit{Inv}_i, \mathit{Grd}_i)\; \forall\, i \in I\}\,. \tag{21}$$

Correspondingly, the external behaviour of the plant is seen to be

$$\mathfrak{B}_{ex} := \{((u_i)_{i\in I}, (y_i)_{i\in I})|\; \exists\, (x_i)_{i\in I} : ((u_i)_{i\in I}, (y_i)_{i\in I}, (x_i)_{i\in I}) \in \mathfrak{B}\}\,. \tag{22}$$

Let (Z_i, W_i, δ_i), $i \in I$, denote the finite automaton estimating $(\Phi_i,\, Inv_i,\, Grd_i)$ as defined in Section 4. Further, let $\mathfrak{B}_{ex}(Z_i,\, W_i,\, \delta_i)$, $i \in I$, denote the external behaviour induced by $(Z_i,\, W_i,\, \delta_i)$. Then

$$\mathfrak{B}_{ca} := \{((u_i)_{i\in I},\, (y_i)_{i\in I})| \\ (u_i,\, y_i) \in \mathfrak{B}_{ex}(Z_i,\, W_i,\, \delta_i)\ \forall\, i \in I\} \qquad (23)$$

forms the external behaviour when running all automata $(Z_i,\, W_i,\, \delta_i)$ synchronised by the $Tick$ event. From equation (20) it follows

$$\mathfrak{B}_{ca} \supseteq \mathfrak{B}_{ex}\,, \qquad (24)$$

i.e. the composition of the individual conservative approximations form a conservative approximation of the multiple switched flow system w.r.t. the external behaviour.

Note that the “set product like composition” as it is done in equations (21) and (23) will make no sense at all when applied to logic time behaviours. This emphasizes that retention of timing aspects is indeed crucial. However, after composing the individual approximations we may discard the timing information in order to run a synthesis procedure from DES theory based on logic time. This is also the framework in which our contribution [9, 11] is settled, providing an approximation based approach that is now applicable to multiple switched flow systems by referring to equation (24).

6 Conclusions

In this contribution, we provide an approximation scheme for multiple switched flow systems that is suitable for approximation based supervisory control synthesis The crucial feature is the retention of timing aspects when *first*, approximating individual sub-systems separately. It then becomes rather straightforward to *second*, compose the individual automata into an approximation of a multiple switched flow system. As the approximation of switched flow systems is of high order complexity w.r.t. the state space dimension, our overall procedure is expected to save a reasonable amount of computation time when compared to *first*, combining the switched flow systems and *second*, running an approximation. Future work will investigate distributed or decentralized control applied to individual switched flow systems based on the proposed approximation procedure.

References

[1] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T. A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J. and Yovine, S. 1995. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, vol. 138, pp. 3–34.

[2] Brandin, B. A. and Wonham, W. M. 1994. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 329–342.

[3] Bournez, O., Maler, O. and Pnueli, A. 1999. Orthogonal polyhedra: representation and computation. In Vaandrage, F. and van Schuppen, J. (eds.) *Hybrid Systems: Computation and Control, Lecture Notes in Computer Sciences*, vol. 1569, pp. 46-60, Springer-Verlag.

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

[5] Chutinan, A. and Krogh, B. H. 1999. Computing Approximating Automata for a Class of Linear Hybrid Systems. In Antsaklis, P. J., Kohn, W., Lemmon, M., Nerode, A. and Sastry, S. (eds.) *Hybrid Systems V, Lecture Notes in Computer Sciences*, vol. 1567, pp. 16–38, Springer-Verlag.

[6] Cury, J. E. R., Krogh, B. A. and Niinomi, T. 1998. Synthesis of supervisory controllers for hybrid systems based on approximating automata. In Antsaklis, P. J. and Nerode, A. (eds.) *IEEE Transactions on Automatic Control*, vol 43, no 4, Special issue on hybrid systems, pp. 564-568.

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

[8] Kurzhanski, A. B. and Varaiya, P. 2000. Ellipsoidal techniques for reachability analysis. In Lynch, N. and Krogh, B. (eds.) *Hybrd Systems: Computation and Control, Lecture Notes in Computer Sciences*, vol. 1790, pp. 203–213, Springer-Verlag.

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

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

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

[12] Raisch, J. and O'Young, S. D. 1997. A totally ordered set of discrete abstractions for a given hybrid or continuous system. In Antsaklis, P. J., Kohn, W., Nerode, A. and Sastry, S. (eds.) *Hybrid Systems IV, Lecture Notes in Computer Sciences*, vol. 1273, pp. 342–360, Springer-Verlag.

[13] Raisch, J. and O'Young, S. D. 1998. Discrete approximation and supervisory control of continuous systems. In Antsaklis, P. J. and Nerode, A. (eds.) *IEEE Transactions on Automatic Control*, vol 43, no 4, Special issue on hybrid systems, pp. 569–573.

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

[15] 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.

[16] 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.