Control Synthesis of Nonlinear Sampled Switched Systems using Euler's Method

SNR'17, Uppsala

Adrien Le Coënt[1], Florian De Vuyst[1],
Ludovic Chamoin[2], Laurent Fribourg[3]

April 22, 2017

[1]CMLA - ENS Cachan
[2]LMT - ENS Cachan
[3]LSV - CNRS - ENS Cachan

Introduction

Framework

- Framework of the switched control systems: one selects the working modes of the system over time, every mode is described by ordinary differential equations (ODEs)
- The tools we used:
  - Tiling (space discretization)
  - Zonotopes (symbolic representation of sets)
  - Validated simulation
- The tools we introduce:
  - A new error bound for the explicit Euler scheme
  - Associated set based computations using balls

Outline

Outline

Switched systems

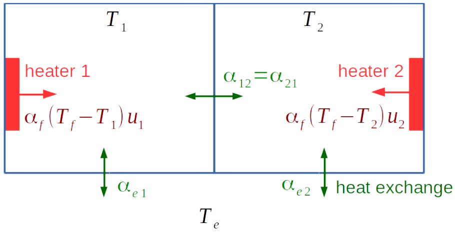
A continuous switched system

$$\dot{x}(t) = f_{\sigma(t)}(x(t))$$

- state $x(t) \in \mathbb{R}^n$
- finite set of (switched) modes $U = \{1, \ldots, N\}$
- state dependent rule σ which associates a mode $u \in U$ to a state $x(t)$

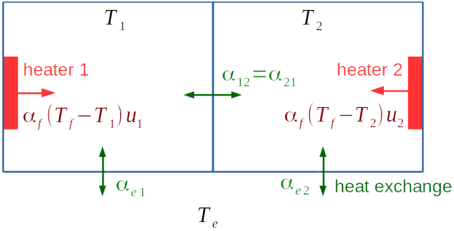
We focus on sampled switched systems:
given a sampling period $\tau > 0$, switchings will occur at times τ, 2τ, ...

Example: Two-room apartment



$$\begin{pmatrix}\dot{T}_1\\ \dot{T}_2\end{pmatrix} = \begin{pmatrix}-\alpha_{21}-\alpha_{e1}-\alpha_f u_1 & \alpha_{21}\\ \alpha_{12} & -\alpha_{12}-\alpha_{e2}-\alpha_f u_2\end{pmatrix}\begin{pmatrix}T_1\\ T_2\end{pmatrix}+\begin{pmatrix}\alpha_{e1}T_e+\alpha_f T_f u_1\\ \alpha_{e2}T_e+\alpha_f T_f u_2\end{pmatrix}.$$

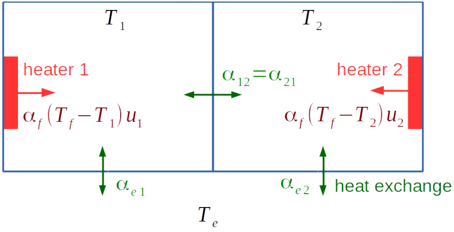
Example: Two-room apartment



$$\begin{pmatrix}\dot{T}_1\\ T_2\end{pmatrix}=\begin{pmatrix}-\alpha_{21}-\alpha_{e1}-\alpha_f u_1 & \alpha_{21}\\ \alpha_{12} & -\alpha_{12}-\alpha_{e2}-\alpha_f u_2\end{pmatrix}\begin{pmatrix}T_1\\ T_2\end{pmatrix}+\begin{pmatrix}\alpha_{e1}T_e+\alpha_f T_f u_1\\ \alpha_{e2}T_e+\alpha_f T_f u_2\end{pmatrix}.$$

- Modes: $\begin{pmatrix}u_1\\ u_2\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix},\begin{pmatrix}0\\ 1\end{pmatrix},\begin{pmatrix}1\\ 0\end{pmatrix},\begin{pmatrix}1\\ 1\end{pmatrix}$; sampling period τ

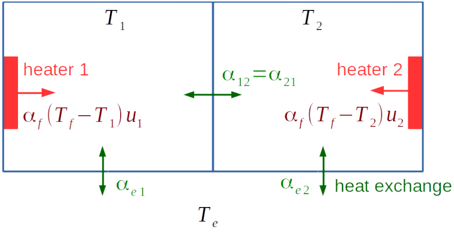
Example: Two-room apartment



$$T_1(t+\tau) = f_1(T_1(t), T_2(t), u_1)$$
$$T_2(t+\tau) = f_2(T_1(t), T_2(t), u_2)$$

- Modes: $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$; sampling period τ

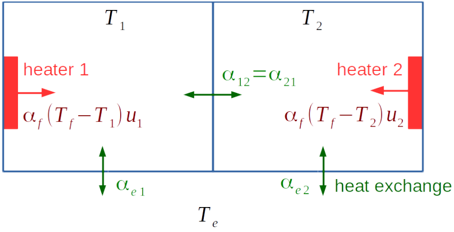
Example: Two-room apartment



$$T_1(t+\tau) = f_1(T_1(t), T_2(t), u_1)$$
$$T_2(t+\tau) = f_2(T_1(t), T_2(t), u_2)$$

- Modes: $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$; sampling period τ
- A pattern π is a finite sequence of modes, e.g. $\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$

Example: Two-room apartment



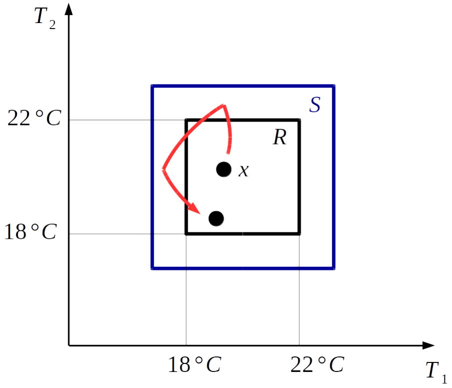
$$T_1(t+\tau) = f_1(T_1(t), T_2(t), u_1)$$
$$T_2(t+\tau) = f_2(T_1(t), T_2(t), u_2)$$

- Modes: $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$; sampling period τ
- A pattern π is a finite sequence of modes, e.g. $\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$
- A state dependent control consists in selecting at each τ a mode (or a pattern) according to the current value of the state.

Control Synthesis Problem

Being given a recurrence set R and a safety set S, we consider the state-dependent control problem of synthesizing σ:
At each sampling time t, find the appropriate switched mode $u \in U$ according to the current value of x, such that

Recurrence: after some steps of time, the state returns into R with safety in S



Outline

Numerical integration, reachability analysis

- Classical (non guaranteed) methods: Euler, Runge-Kutta, implicit, explicit schemes...
- Guaranteed reachability analysis: Enclosing solutions, error bounding, additional hypotheses
- State-of-the-art:
  - Monotone, ISS, incrementally stable systems [Girard, Sontag, Zamani, Tabuada...]
  - Reachability analysis using zonotopes [Dang, Girard, Althoff...]
  - Validated simulation, guaranteed integration [Moore, Lohner, Bertz, Makino, Nedialkov, Jackson, Corliss, Chen, Ábrahám, Sankaranarayanan, Taha, Chapoutot,...]
  - Ellipsoid methods [Kurzhanski, Varaiya, Dang...]
  - Sensitivity Analysis [Donzé, Maler...]

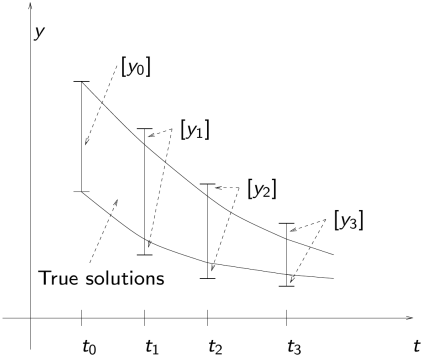
Validated Numerical Methods for IVPs for ODEs

[Moore, 1966]

For $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, we consider the IVP for ODEs

$$\dot{y}(t) = f(y(t))$$

We denote the solution by $y(t, t_0, y_0)$.
For an interval $[y_0]$, $y(t; t_0, [y_0]) = \{y(t; t_0, y_0) | y_0 \in [y_0]\}$.



Our goal is to find interval vectors $[y_j]$ that are guaranteed to contain $y(t; t_0, [y_0])$ at the points $t_0 < t_1 < \cdots < t_N = \tau$.

Taylor Series Methods for IVPs for ODEs

[Moore, 1966]

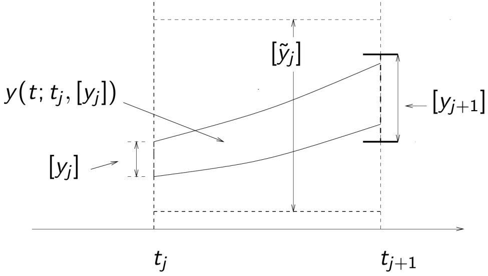
Suppose that we have computed $[y_j]$ at t_j, such that $y(t_j; t_0, [y_0]) \subseteq [y_j]$.
We advance the solution by using two algorithms.

Algorithm I

- validates existence and uniqueness of the solution $[t_j, t_[j + 1]$
- computes an a priori enclosure $[\tilde{y}_j]$ such that

$$y(t; t_j, [y_j]) \subseteq [\tilde{y}_j] \quad \text{for } t \in [t_j, t_{j+1}].$$

Algorithm II computes a tighter enclosure $[y_{j+1}]$ of $y(t_{j+1}; t_0, [y_0])$.



Outline

Hypotheses

$(H0)$ (Lipschitz): for all $j \in U$, there exists a constant $L_j > 0$ such that:

$$\|f_j(y) - f_j(x)\| \leq L_j \|y - x\| \quad \forall x, y \in S.$$

$(H1)$ (One-sided Lipschitz/Strong monotony): for all $j \in U$, there exists a constant $\lambda_j \in \mathbb{R}$ such that

$$\langle f_j(y) - f_j(x), y - x \rangle \leq \lambda_j \|y - x\|^2 \quad \forall x, y \in T,$$

Let us define the constants: C_j for all $j \in U$:

$$C_j = \sup_{x \in S} \; L_j \|f_j(x)\| \quad \text{for all} \quad j \in U.$$

NB: constants computed by constrained optimization.

Computation of the constants

Computation of L_j, C_j, λ_j ($j \in U$) realized with constrained optimization algorithms, applied on the following optimization problems:

- Constant L_j:

$$L_j = \sup_{x,y \in S,\ x \neq y} \frac{\|f_j(y) - f_j(x)\|}{\|y - x\|}$$

- Constant C_j:

$$C_j = \sup_{x \in S} L_j \|f_j(x)\|$$

- Constant λ_j:

$$\lambda_j = \sup_{x,y \in T,\ x \neq y} \frac{\langle f_j(y) - f_j(x), y - x\rangle}{\|y - x\|^2}$$

Notations

We will denote by $\phi_j(t; x^0)$ the solution at time t of the system:

$$\dot{x}(t) = f_j(x(t)),$$
$$x(0) = x^0.$$

Given an initial point $\tilde{x}^0 \in S$ and a mode $j \in U$, we define the following "linear approximate solution" $\tilde{\phi}_j(t; \tilde{x}^0)$ for t on $[0, \tau]$ by:

$$\tilde{\phi}_j(t; \tilde{x}^0) = \tilde{x}^0 + t f_j(\tilde{x}^0).$$

NB: approximation by forward Euler scheme

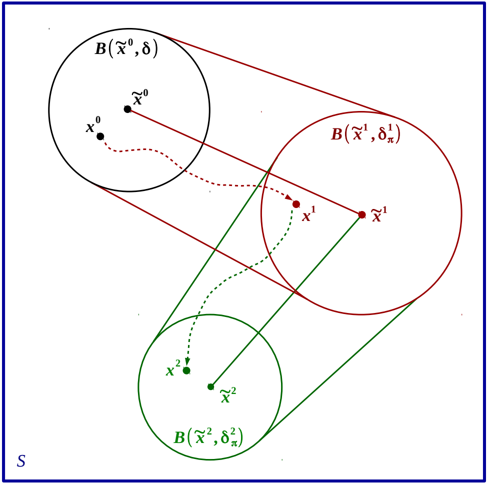
Main result

Theorem

Given a sampled switched system satisfying (H0-H1), consider a point $\tilde{x}^0$ *and a positive real* δ. *We have, for all* $x^0 \in B(\tilde{x}^0, \delta)$, $t \in [0, \tau]$ *and* $j \in U$*:*
$\phi_j(t; x^0) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t))$.
with

- *if* $\lambda_j < 0$: $\delta_j(t) = \left(\delta^2 e^{\lambda_j t} + \frac{C_j^2}{\lambda_j^2}\left(t^2 + \frac{2t}{\lambda_j} + \frac{2}{\lambda_j^2}\left(1 - e^{\lambda_j t}\right)\right)\right)^{\frac{1}{2}}$
- *if* $\lambda_j = 0$: $\delta_j(t) = \left(\delta^2 e^t + C_j^2(-t^2 - 2t + 2(e^t - 1))\right)^{\frac{1}{2}}$
- *if* $\lambda_j > 0$: $\delta_j(t) = \left(\delta^2 e^{3\lambda_j t} + \frac{C_j^2}{3\lambda_j^2}\left(-t^2 - \frac{2t}{3\lambda_j} + \frac{2}{9\lambda_j^2}\left(e^{3\lambda_j t} - 1\right)\right)\right)^{\frac{1}{2}}$

Application to guaranteed integration



Sketch of the proof

Error equation

$$\frac{d}{dt}(x(t) - \tilde{x}(t)) = \left(f_j(x(t)) - f_j(\tilde{x}^0)\right),$$

Transformation into a differential inequality

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}(\|x(t) - \tilde{x}(t)\|^2) &= \left\langle f_j(x(t)) - f_j(\tilde{x}^0), x(t) - \tilde{x}(t)\right\rangle \\
&\leq \langle f_j(x(t)) - f_j(\tilde{x}(t)), x(t) - \tilde{x}(t)\rangle + \\
&\qquad \|f_j(\tilde{x}(t)) - f_j(\tilde{x}^0)\|\|x(t) - \tilde{x}(t)\| \\
&\leq \lambda_j\|x(t) - \tilde{x}(t)\|^2 + L_j t\,\|f_j(\tilde{x}^0)\|\,\|x(t) - \tilde{x}(t)\|
\end{aligned}$$

Then integration of the differential inequality, knowing that $\|x(t) - \tilde{x}(t)\| \leq \frac{1}{2}\left(\alpha\|x(t) - \tilde{x}(t)\|^2 + \frac{1}{\alpha}\right)$ for $\alpha > 0$

Application to guaranteed integration

Given a sampled switched system satisfying (H0-H1), consider a point $\tilde{x}^0 \in S$, a real $\delta > 0$ and a mode $j \in U$ such that:

1. $B(\tilde{x}^0, \delta) \subseteq S$,
2. $B(\tilde{\phi}_j(\tau; \tilde{x}^0), \delta_j(\tau)) \subseteq S$, and
3. $\frac{d^2(\delta_j(t))}{dt^2} > 0$ for all $t \in [0, \tau]$.

Then we have, for all $x^0 \in B(\tilde{x}^0, \delta)$ and $t \in [0, \tau]$: $\phi_j(t; x^0) \in S$.

Convexity of the trajectories

Example of a DC-DC converter:

The dynamics is given by the equation $\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}$ with $\sigma(t) \in U = \{1, 2\}$. The two modes are given by the matrices:

$$A_1 = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} \quad B_1 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

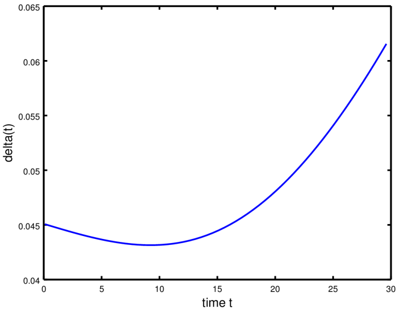
$$A_2 = \begin{pmatrix} -\frac{1}{x_l}(r_l + \frac{r_0.r_c}{r_0+r_c}) & -\frac{1}{x_l}\frac{r_0}{r_0+r_c} \\ \frac{1}{x_c}\frac{r_0}{r_0+r_c} & -\frac{1}{x_c}\frac{r_0}{r_0+r_c} \end{pmatrix} \quad B_2 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

with $x_c = 70$, $x_l = 3$, $r_c = 0.005$, $r_l = 0.05$, $r_0 = 1$, $v_s = 1$.

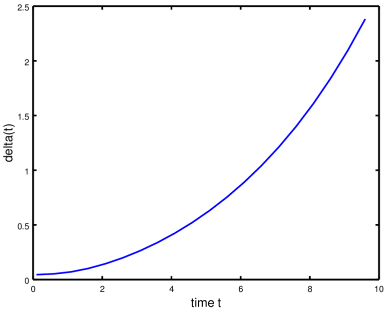
λ_1	-0.014215
λ_2	0.142474
C_1	6.7126×10^{-5}
C_2	2.6229×10^{-2}

Convexity of the trajectories

Example of a DC-DC converter:

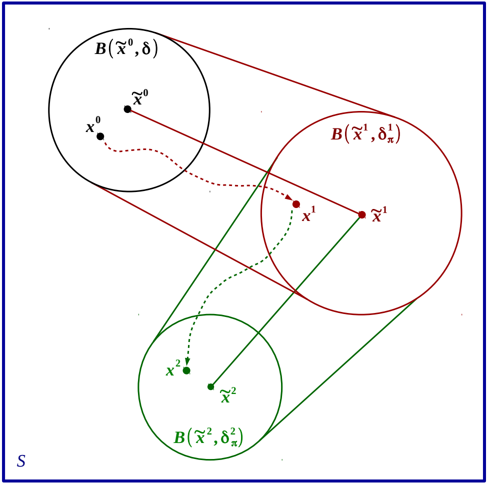


$\lambda_1 = -0.014215$



$\lambda_2 = 0.142474$

Application to guaranteed integration

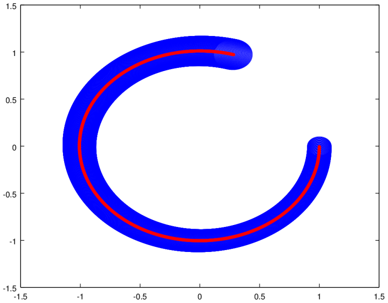


Application to guaranteed integration

A simple rotation:

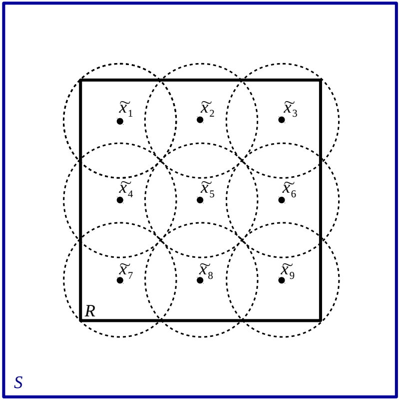
$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x$$

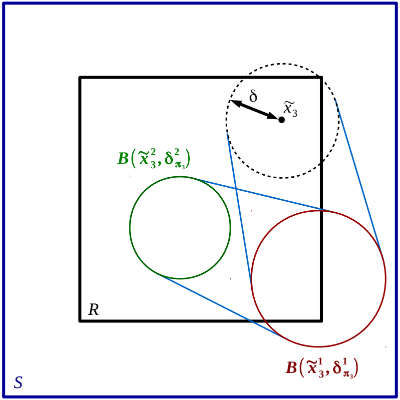
We have: $\lambda = 0$, $C = 4.2$, $L = 1$
Initial radius: 0.1
Time step: 0.005



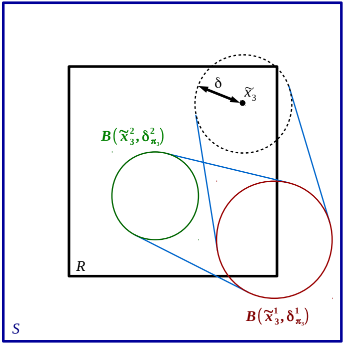
Outline

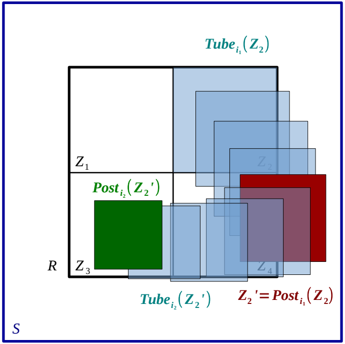
Control synthesis





Validated simulation vs Euler





Building ventilation

[Meyer, Nazarpour, Girard, Witrant, 2014]

Dynamics of a four-room apartment:

$$\frac{dT_i}{dt} = \sum_{j \in \mathcal{N}^*} a_{ij}(T_j - T_i) + \delta_{s_i} b_i (T_{s_i}^4 - T_i^4) + c_i \max\left(0, \frac{V_i - V_i^*}{\bar{V}_i - V_i^*}\right)(T_u - T_i).$$

$\mathcal{N}^* = \{1, 2, 3, 4, u, o, c\}$

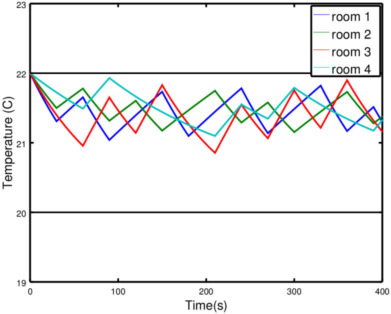
Control inputs: V_1 and V_4 can take the values 0V or 3.5V, and V_2 and V_3 can take the values 0V or 3V

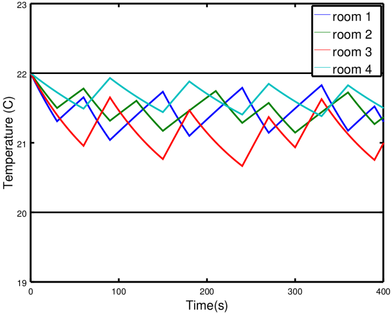
$\Rightarrow$ 16 switching modes

Building ventilation

	Euler	DynIBEX
R S	$[20, 22]^4$ $[19, 23]^4$	
τ	30	
Time subsampling	No	
Complete control	Yes	Yes
$\max_{j=1,...,16} \lambda_j$ $\max_{j=1,...,16} C_j$	-6.30×10^{-3} 4.18×10^{-6}	
Number of balls/tiles Pattern length	4096 1	252 1
CPU time	63 seconds	249 seconds

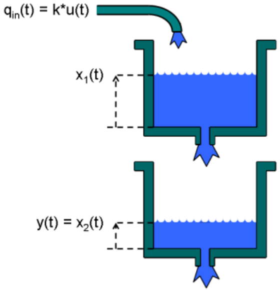
Building ventilation





Two-tank system

The behavior of x_1 is given by $\dot{x}_1 = -x_1 - 2$ when the tank 1 valve is closed, and $\dot{x}_1 = -x_1 + 3$ when it is open. Likewise, x_2 is driven by $\dot{x}_2 = x_1$ when the tank 2 valve is closed and $\dot{x}_2 = x_1 - x_2 - 5$ when it is open.



Two-tank system

	Euler	DynIBEX
R S	$[-1.5, 2.5] \times [-0.5, 1.5]$ $[-3, 3] \times [-3, 3]$	
τ	0.2	
Time subsampling	$\tau/10$	
Complete control	Yes	Yes
λ_1 λ_2 λ_3 λ_4 C_1 C_2 C_3 C_4	0.20711 -0.50000 0.20711 -0.50000 11.662 28.917 13.416 32.804	
Number of balls/tiles Pattern length	64 6	10 6
CPU time	58 seconds	246 seconds

Comparison with state-of-the-art

	Euler	DynIbex
Lotka-Volterra	fail	✓
Van der Pol	fail	✓
DC-DC	fail	✓
4-room	✓	✓
2-tank	✓	✓
helicopter	✓	✓

Advantages and limits

- Advantages:
  - Computationally very cheap
  - Only one punctual evaluations of f by step
  - No numerical integration
  - Easy implementation
- Limits:
  - Can lack precision when $\lambda > 0$
  - Can require sub-sampling
  - No perturbations (yet)

Conclusions and future work

Conclusions:

- Guaranteed control of nonlinear switched systems
- Use of a simple forward Euler method
- New error bound
- Easy implementation

Future work:

- Perturbations
- Extension to compositional/distributed synthesis