Recent trends in control synthesis for hybrid systems: a personal view

Laurent Fribourg

LSV – CNRS & ENS Cachan, U. Paris-Saclay

November 4, 2016 - IRT SystemX

Plan

I Classical Control

II Hybrid systems

III Set-based approach

IV (Bi)simulation

V MINIMATOR

VI Compositionality

VII Model reduction

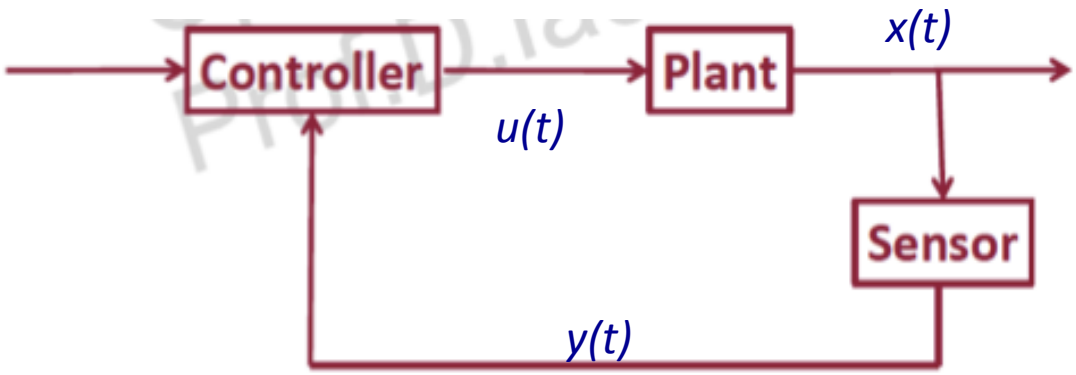
VIII Conclusion

I. Classical control

Schematic view of a control system

- Plant: dynamics with a state variable *x(t)* governed by
  - *dx/dt = f(x,u)* (continuous-time form)
  - *x(t+1) = f(x(t),u(t))* (discrete-time form)

- Sensors: gives a partial information *y(t)* about *x(t)*
- Controller: computes the law *u(t)* as a function of *y(t)*



http://www.dis.uniroma1.it/~iacoviel/

Principle of feedback

- **Feedback:** The actual operation of the control system is compared to the desired operation and the input *u(t)* to the plant is adjusted on the basis of this comparison.

- **Feedback control systems** are able to operate satisfactorily despite adverse conditions, such as disturbances and variations in plant properties

Optimal control (the moon lander)

- Aim: bring a spacecraft to a soft landing on the lunar surface, *using the least amount of fuel*

- The state variable x is a triple (h,v,m) with:
  - $h(t)$ *height* at time t
  - $v(t)$ *velocity* ($= dh/dt$)
  - $m(t)$ *mass* of spacecraft

- The control (or input) $u(t)$ is the *thrust* at time t

http://www.dis.uniroma1.it/~iacoviel/

Moon lander

- Consider Newton's law: $m\, dv/dt = -\ g\, m + u$

  This gives:

  $dv/dt = -\ g + u/m$

  $dh/dt = v$

  $dm/dt = -\ k\, u$

- The problem is to find $u(.)$ in order to

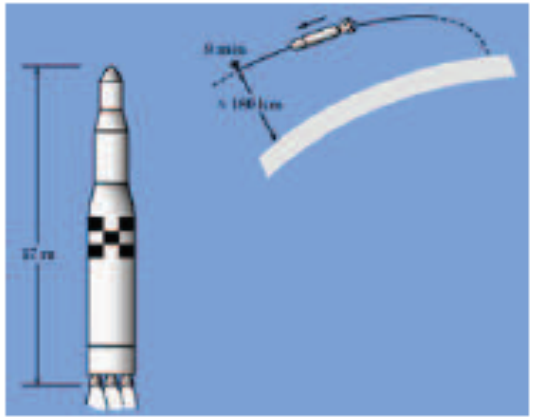
  minimize the amout of fuel, i.e., *maximize* the amount remaining once we landed, i.e., maximize J defined by:

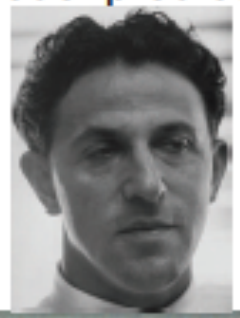
  $$J(u(.))=m(T)$$

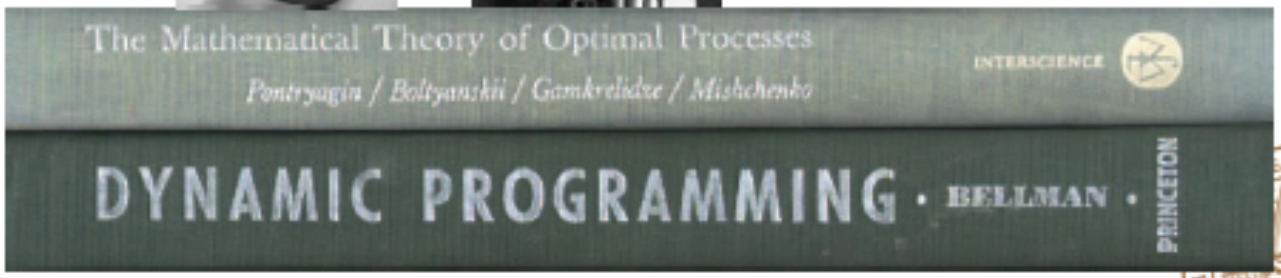
  where T is the *first time*: $h(T)=0,\ v(T)=0$.

Optimal Control

- Hamilton, Jacobi, Bellman 1957
- Euler, Lagrange, Pontryagin 1962
- Model predictive control







History of Control – The Second Wave

http://www.control.lth.se/Staff/KarlJohanAstrom.html

Limit: scalability

- The real subsystems are often numerous:
  - Multi variable
  - High dimension
  - Nonlinear
  - Time-Varying
  - Poorly modelled

- Thus they are often outside the bounds of existing classical theory, and/or existing computational tools

*www.**ifac**-**control**.org/about/**ifac**-**50**-lectures/2-Anderson.pdf*

Example

- The pitch control system on a commercial aircraft (2006) has two inputs, two outputs, stochastic disturbance, is open loop unstable. The state dimension is about *50*.

- Challenge: How to design a (low order) controller

*www.**ifac**-**control**.org/about/**ifac**-**50**-lectures/2-Anderson.pdf*

New control problems

- Digital computer as a control system component:

  → hybrid system (with switch control)
  → (new) *combinatorial explosion*:
  nb of possible switches grows exponential with time horizon

- Provable safe design

- Complex networked systems

- Sensor and actuator rich systems

- Autonomous distributed systems

http://www.control.lth.se/Staff/KarlJohanAstrom.html

II. Hybrid systems

The discrete-time dynamics of a hybrid system is

$$x(t+1)=f(x(t),u(t))$$

where *u(t) is a discrete variable* that takes its values on a finite domain *U*, eg: *{0,1}* (instead of a dense domain, eg: *[0,1]*)

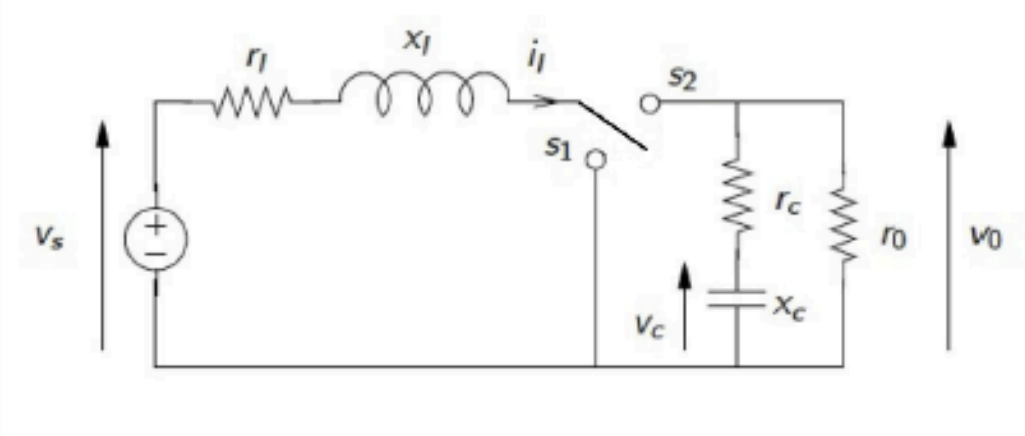
The control synthesis problem consists in choosing at each *t = 0,1,2,…* a mode (value of *u*) according to the current value of *x* (or observation *y*) in order to meet a temporal property *spec(x)*

A special class: switched systems

- A state variable X
- A set of p modes $U = \{1, 2, \cdots, p\}$
- Each mode $u \in U$ is associated to a dynamic $\dot{X} = f_u(X)$
- Switching modes can only occur at $t = \tau, 2\tau, \cdots$
- Restriction: $\forall u \in U,\ \exists A_u, b_u :\ f_u(X) = A_u X + b_u$

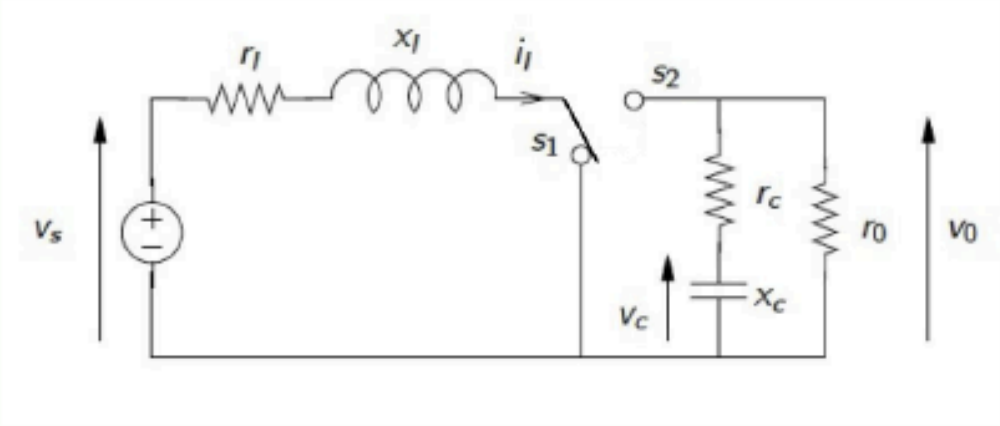
ex: DC-DC converter

Example: Boost DC-DC Converter



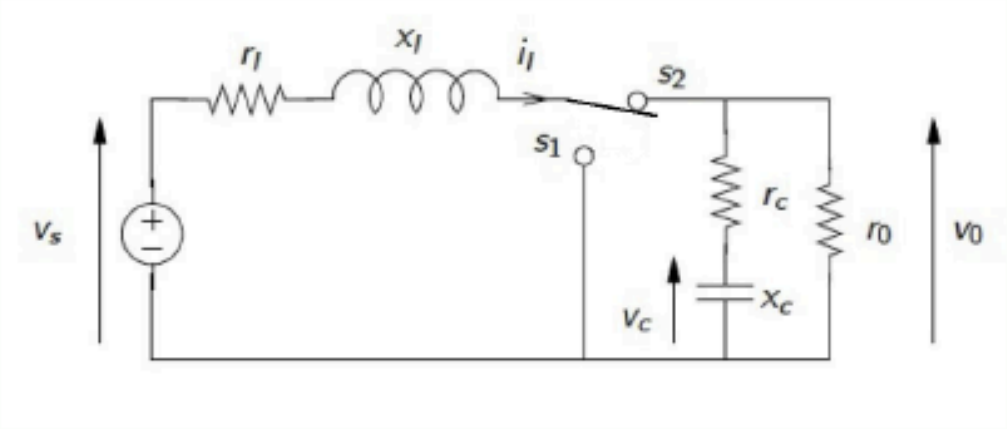
- A state variable $X = (i_l, v_c)^\top$

Example: Boost DC-DC Converter



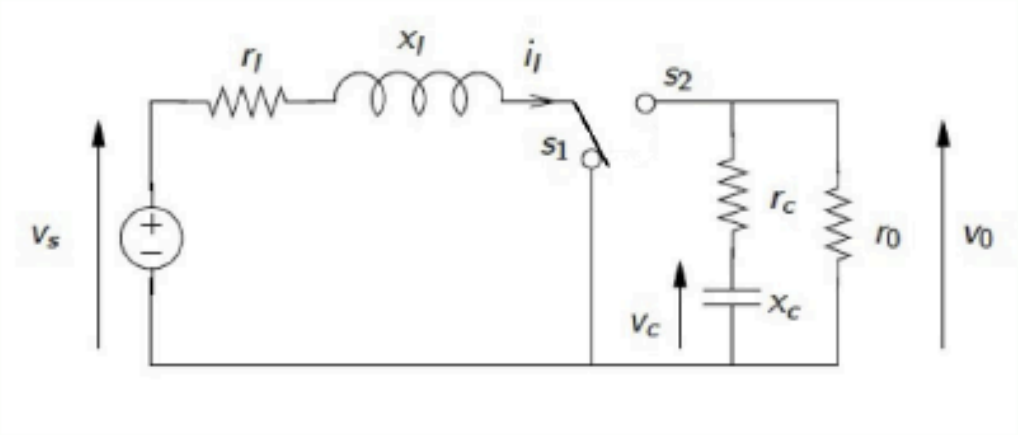
- A state variable $X = (i_l, v_c)^\top$
- 2 possible modes $U = \{1,$

Example: Boost DC-DC Converter



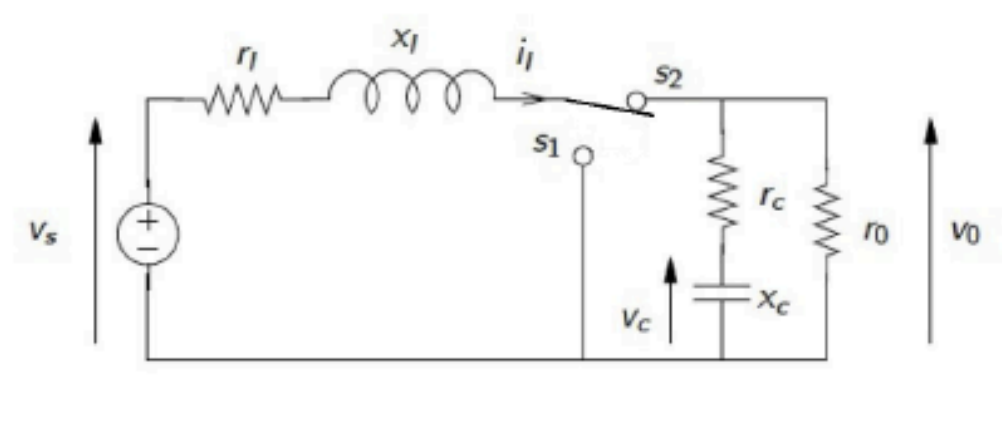
- A state variable $X = (i_l, v_c)^\top$
- 2 possible modes $U = \{1,2\}$

Example: Boost DC-DC Converter



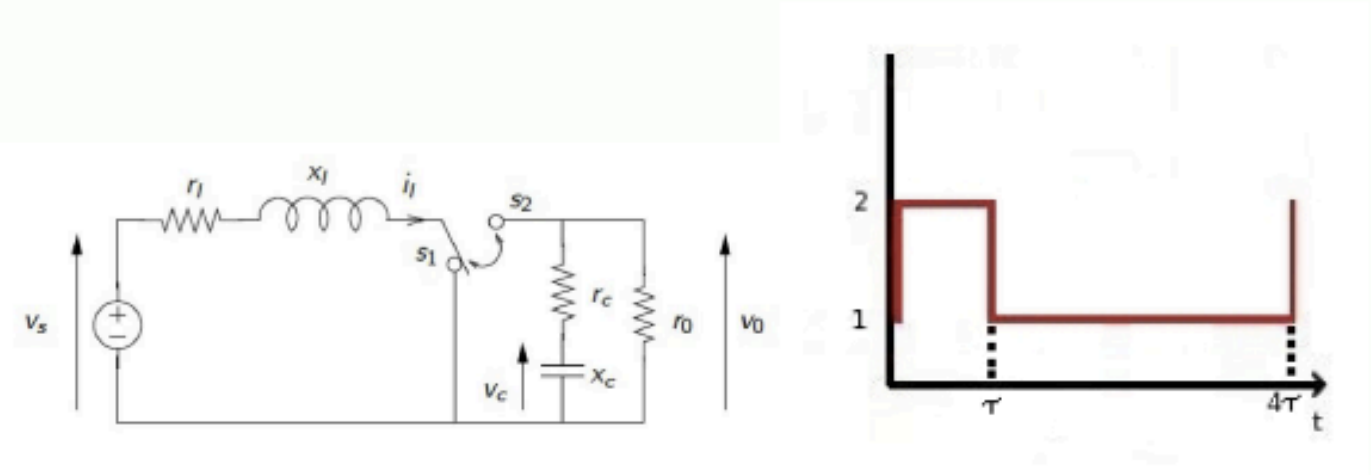
- A state variable $X = (i_l, v_c)^\top$
- 2 possible modes $U = \{1,2\}$
- $\dot{X} = f_1(X) = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} X + \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$

Example: Boost DC-DC Converter



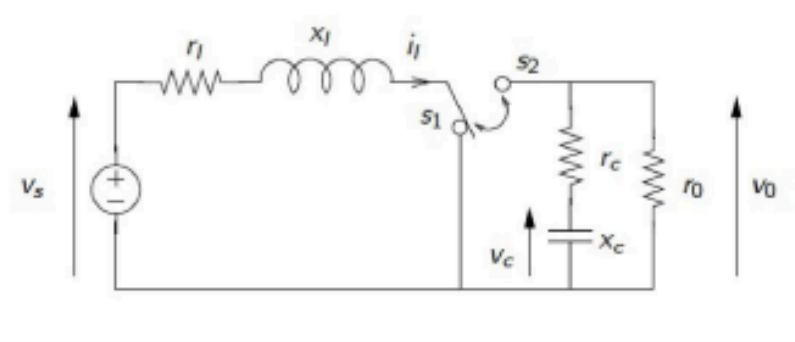
- A state variable $X = (i_l, v_c)^\top$
- 2 possible modes $U = \{1,2\}$
- $\dot{X} = f_1(X) = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} X + \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$
- $\dot{X} = f_2(X) = \begin{pmatrix} -\frac{1}{x_l}(r_l + \frac{r_0.r_c}{r_0+r_c}) & -\frac{1}{x_l}\frac{r_0}{r_0+r_c} \\ \frac{1}{x_c}\frac{r_0}{r_0+r_c} & -\frac{1}{x_c}\frac{1}{r_0+r_c} \end{pmatrix} X + \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$

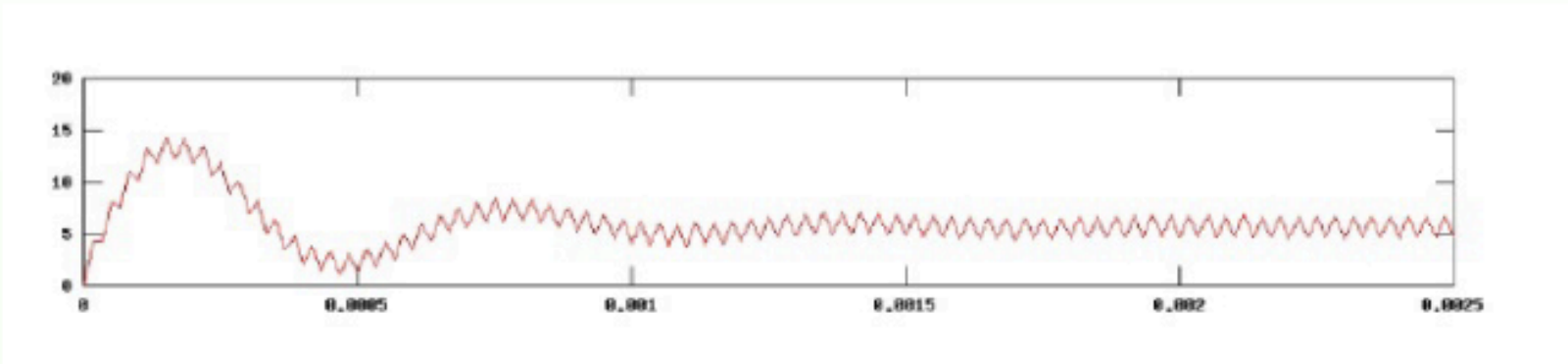
Example: DC-DC Converter



- Modes: $p = 1, 2$; sampling period τ
- A pattern π is a finite sequence of modes (e.g. $(2 \cdot 1 \cdot 1 \cdot 1)$)
- A state dependent control consists to select at each τ a mode (or a pattern) according to the current value X of the state.

Control Objectives (DC-DC Converter Example)





- 1st objective (stability): output voltage regulation around constant desired reference
- 2nd objective (safety) : while maintaining some constraints of current limitation and/or maximal current and voltage ripple

Safety and Stability Properties for the DC-DC Converter

- Example of safety property to be checked: no saturation

$$\forall t \geq 0 : \qquad i_l(t) \leq M$$

- Example of stability property to be checked: voltage regulation

$$|v_{output}(t) - v_{reference}| \leq \varepsilon \text{ for all } t \geq T$$

III Set-based approach

Safety constraint and invariance set

- A safe set *S* is a *constraint* (i.e. a subset of the state space) that should be always satisfied by the state of the system.

- Safety satisfaction can be guaranteed for all time if (and only if) the initial state of the system is contained inside a controlled invariant set of *S*.

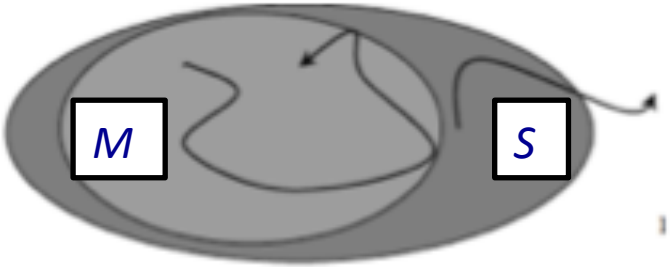
Maximal Controlled Invariant Set

[Bertsekas-Rhodes 1971]

- Def: A subset *X* of *S* is a *controlled invariant subset* of *S* if, for all *x0* in *X*, there is a controlled trajectory issued from *x0* that always stays in *S*

- Prop: The maximal controlled invariant subset (MCIS) *M* of *S* exists.

  Furthermore: *x* in *M* => *f(x,u)* in *M* for some *u*.



Let *X* be a set of states and *u* a mode of *U*, we define the *predecessor* operators:

$Pre_u(X) = \{ x' \mid x = f(x',u) \}$ for some x of X

$Pre(X) = \bigcup_u Pre_u(X) = \{ x' \mid x = f(x',u) \}$ for some x of X, u of U

MCIS algorithm

- Algo

  input: S

  output: M maximal controlled invariant of S

  Initially: $M := S$

  while $Pre(M) \neq M$

  $M := Pre(M) \cap S$

  endwhile

NB1: Algo *terminates* if S finite

NB2: M is the *greatest fixed-point* (*gfp*) of *Pre* contained in S.

Fixed-Points of *Pre*

- **MCIS** of S = *gfp(Pre)* included in S = $\bigcap_k \; Pre^k(S)$

- **Basin of attraction** of S = *lfp(Pre)* containing S = $\bigcup_k \; Pre^k(S)$

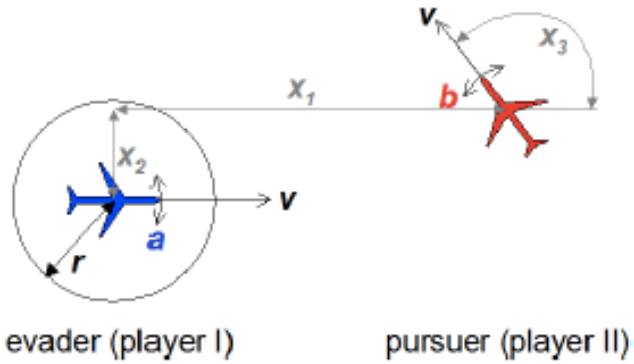
- **Reach-avoid set** of (S, A)

  = set of initial points for which the controlled system reaches S
  while always avoiding A

  = *lfp(Reach-avoid)* containing S = $\bigcup_k \; Reach\text{-}avoid^k(S,A)$

  with $Reach\text{-}avoid(X,A)$ = { x' | x' in $Pre_u(x)$
  & $Pre_u(x) \cap A = \emptyset$ for some control u and x of X }

Application to collision avoidance [Mitchell-Bayen-Tomlin2004]: determination of the unsafe zone using *reach-avoid*

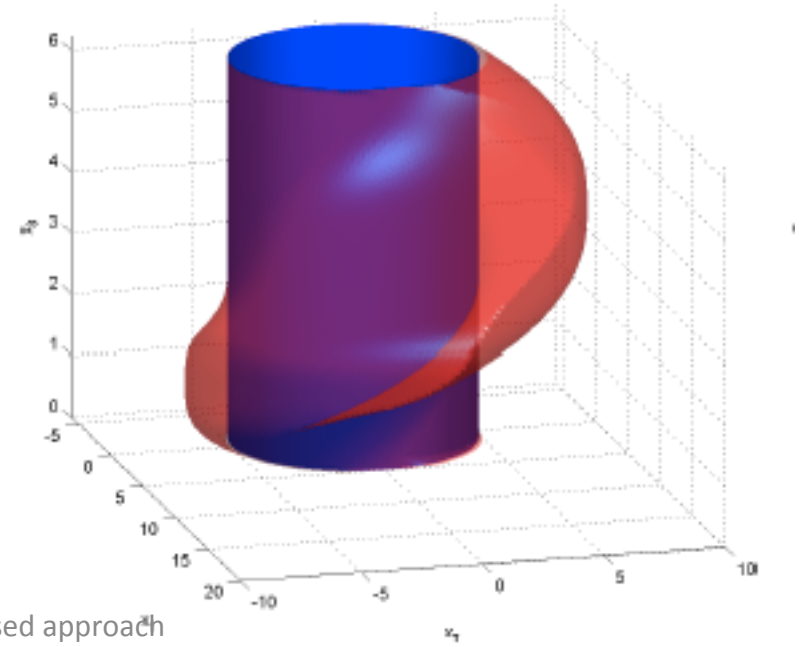


$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v_a + v_b \cos x_3 + a x_2 \\ v_b \sin x_3 - a x_1 \\ b - a \end{bmatrix} = f(x, a, b).$$

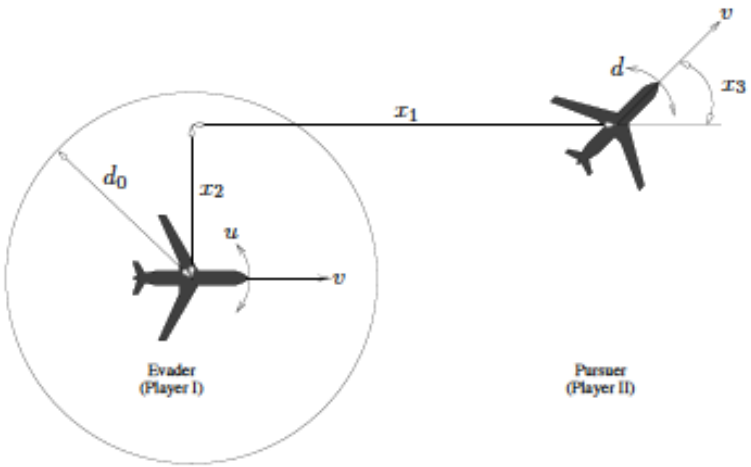
A= { *x1, x2 | x1² + x2² ≤ r²* }

unsafe zone={initial position of red plane (relatively to blue plane) for which there is a risk of collision}

If the red plane is outside the unsafe zone, the blue plane is always ensured to « evade »



Pursuer-evader game [Tomlin-Mitchell-Bayen-Oishi IEEE 2003]



$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v + v\cos x_3 + ux_2 \\ v\sin x_3 - ux_1 \\ d - u \end{bmatrix} = f(x,u,d),$$

Figure 4: Relative coordinate system. Origin is located at the center of the evader.

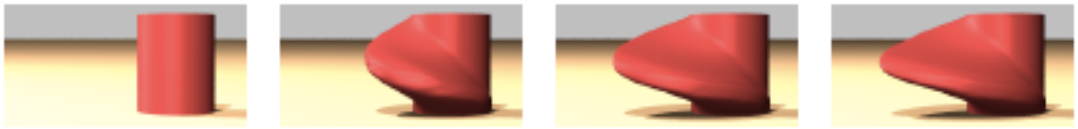


Figure 5: Growth of the reachable set [6] (animation at [60])

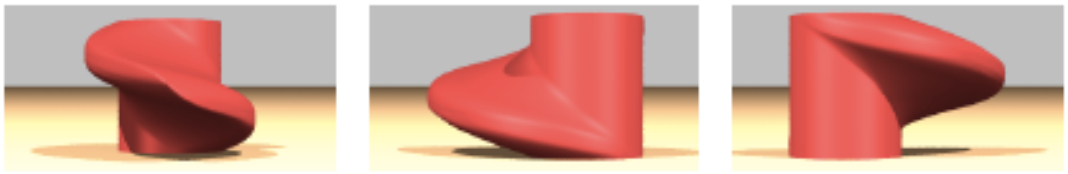
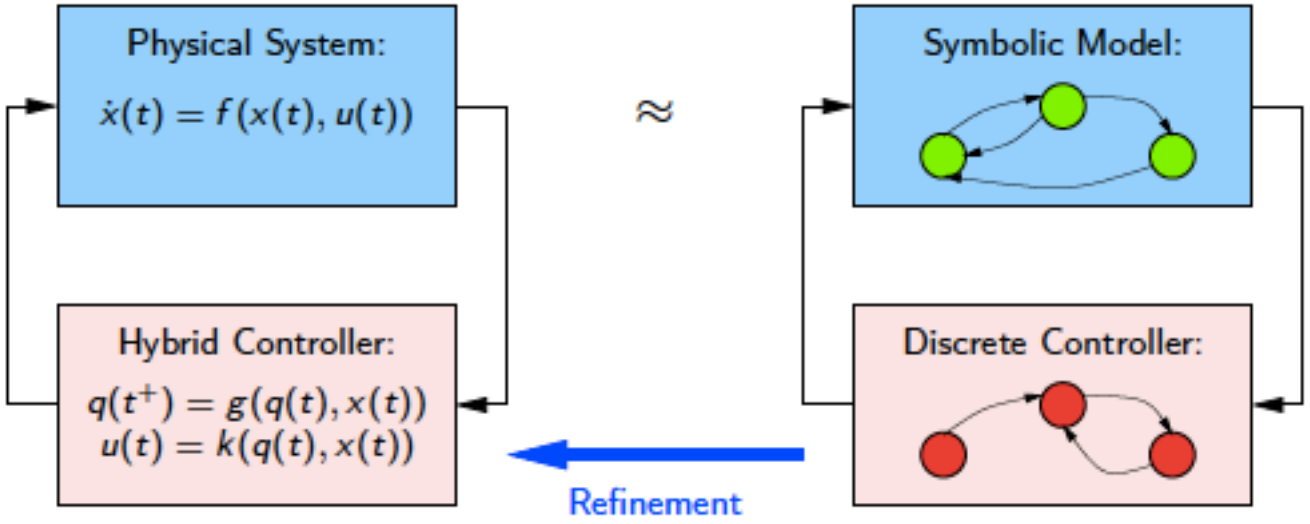


Figure 6: Other views of the reachable set [6] (animation at [60])

IV. Another approach: (bi)simulation

- Pb: non-termination of the fixed-point set calculation in case of infinite state systems

- Idea: find a *bisimular* (≈ equivalent) and finite system

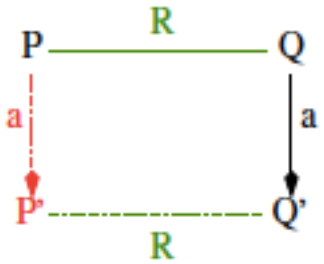


Bisimulation

Consider two transition systems T and T' of state space Σ and Σ' resp.

Def: An *equivalence relation* R on $\Sigma \times \Sigma'$ is a *bi-simulation* if, for all (P,Q) of $\Sigma \times \Sigma'$ with $P\ R\ Q$:

- Any move $Q - a \rightarrow Q'$ of Q can be matched by a move $P - a \rightarrow P'$, with $P'\ R\ Q'$



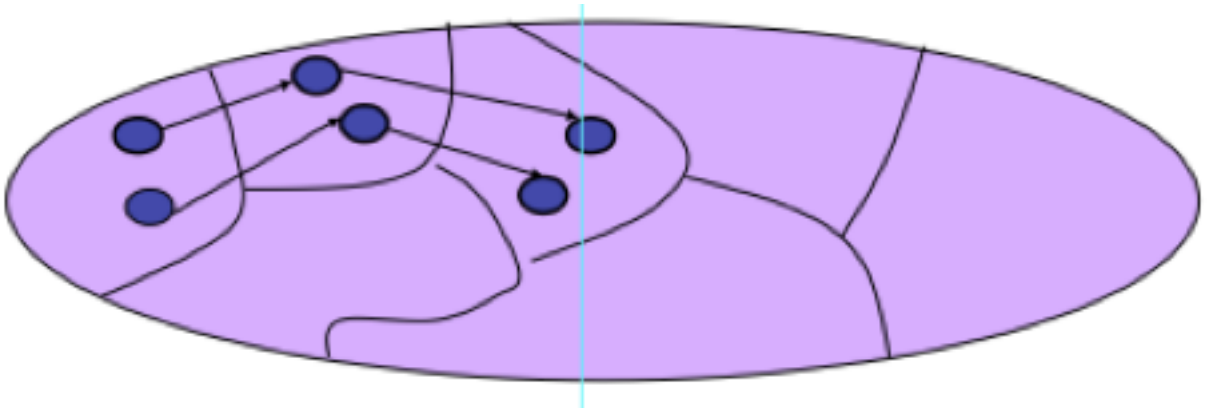
- Conversely, any $P - a \rightarrow P'$ is matched by a move $Q - a \rightarrow Q'$, with $P'\ R\ Q'$

Systems T and T' are said to be *bisimular*

Construction of bisimular quotient automaton

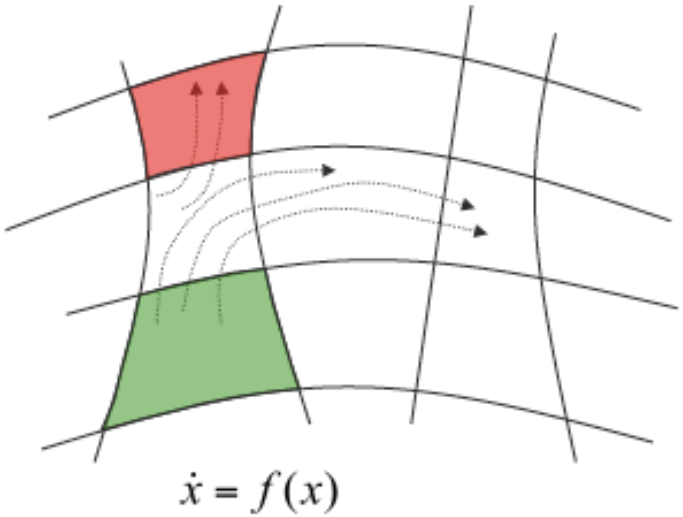
Goal: To partition the infinite state-space of a system *T* into finitely many equivalences classes so that equivalence classes exhibit similar behaviors

(➔ construction of a finite bisimilar quotient automaton *T'*)

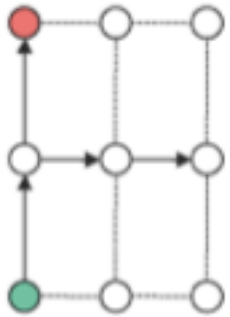


Principle of (bi)simulation

original system *T*



quotient automaton *T'*

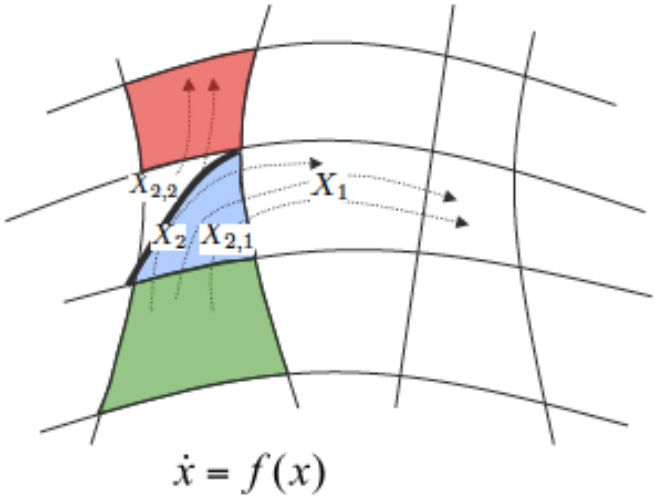


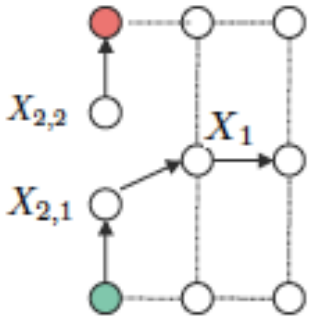
spec: « *there is no trajectory from green to red* »
~(*green* ∧◇*red*) for all trajectories

spec false for this quotient

*https://www.lccc.lth.se/media/LCCC2013/WS1304/**Slides/belta**.pdf*

Quotient refinement



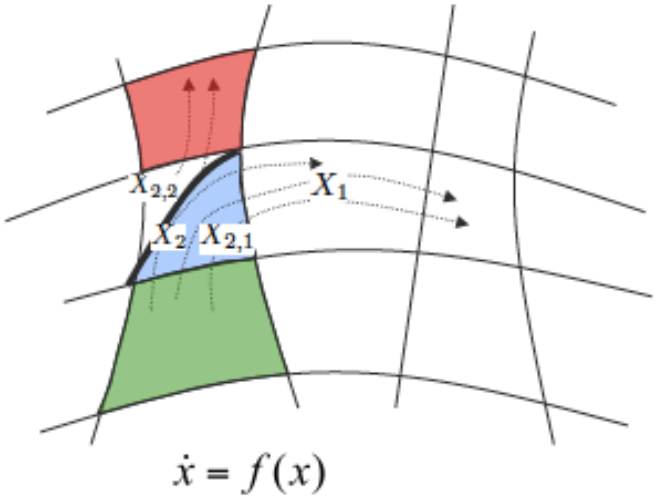


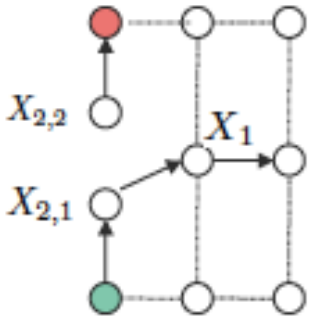
spec. becomes *true* on *T'*

If the quotient *T'* is a *simulation* of *T*, and *spec* is *true* for *T'*
then *spec* is *true* for *T*

NB: *spec* restricted to *universal* properties

Quotient refinement





Partition is refined on Σ using *Pre* operator:

$$X_{2,1} = Pre(X_1) \cap X_2 \neq \emptyset$$

$$X_{2,2} = X_2 \setminus X_{2,1}$$

Bisimulation algorithm

While there exist X_i , X_j such that $\emptyset \subset X_i \cap Pre(X_j) \subset X_i$

$X_{i,1} = X_i \cap Pre(X_j)$

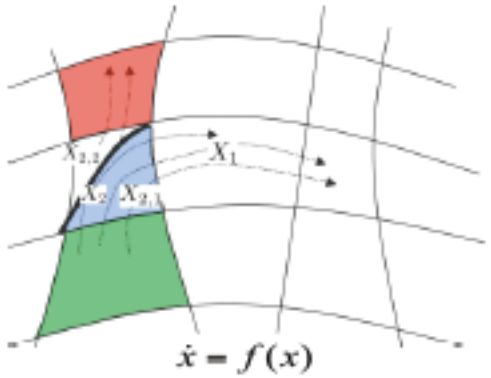
$X_{i,2} = X_i \setminus X_{i,1}$

remove X_i

add $X_{i,1}$, $X_{i,2}$

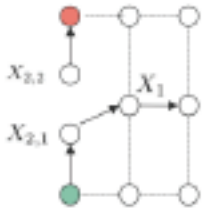
endwhile

A. Bouajjani, J.-C. Fernandez, and N. Halbwachs, 1991.



bisimulation

≡



- If the algorithm terminates,
  the quotient is *finite* and *bisimular* to the original system

- The quotient can be used in lieu of the original system for verification of *spec*

Pb: Unfortunately, termination is rare!

Variant 1 (*spec*-guided refinement)

While there exist X_i , X_j such that $\emptyset \subset X_i \cap Pre(X_j) \subset X_i$

$X_{i,1} = X_i \cap Pre(X_j)$

$X_{i,2} = X_i \setminus X_{i,1}$

remove X_i

add $X_{i,1}$, $X_{i,2}$

construct the quotient

model check the quotient

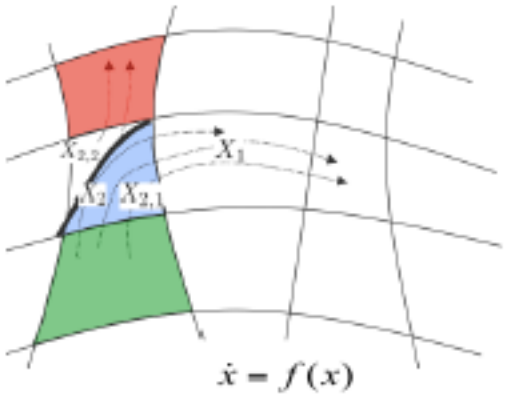
if the spec is satisfied

break

endif

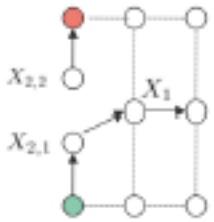
endwhile

A. Chutinan and B. H. Krogh, 2001.



simulation

<



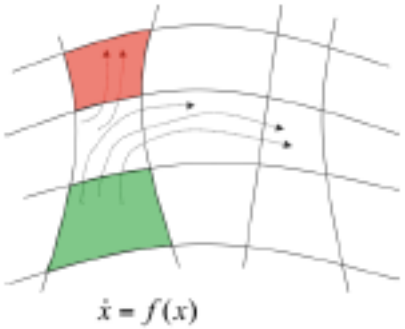
If the algorithm terminates,

the quotient *satisfies spec* and *simulates* the original system (no more bisimulation). The original system is guaranteed to satisfy *spec*.

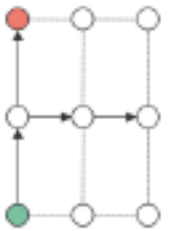
Pb2: Unfortunately, the computation of *Pre* is difficult!

Variant 2 (not using *Pre*)

While TRUE
construct (an over-approximation of) the quotient
model check the quotient
if the spec is satisfied
break;
endif
refine (using some
partitioning scheme)
endwhile



simulation
<



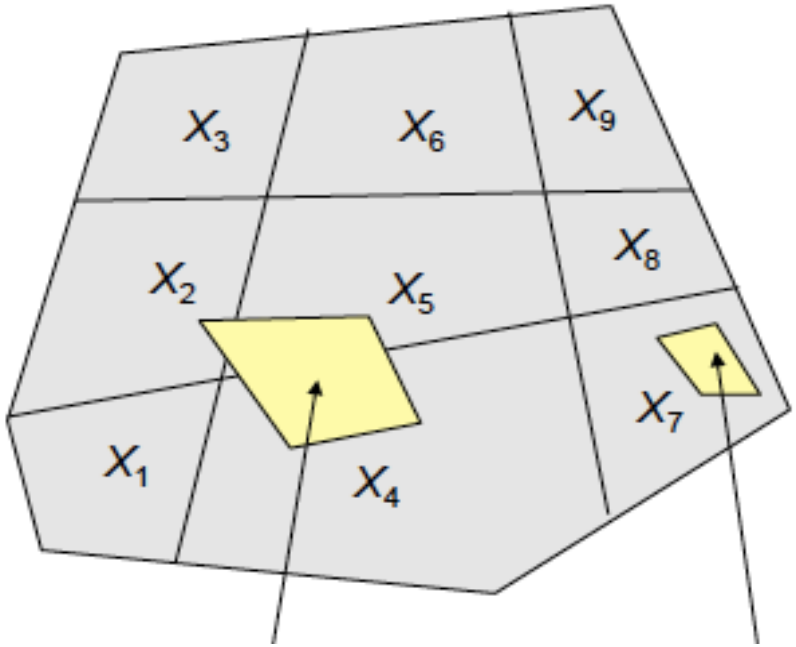
If the algorithm terminates, the original system satisfies *spec*.

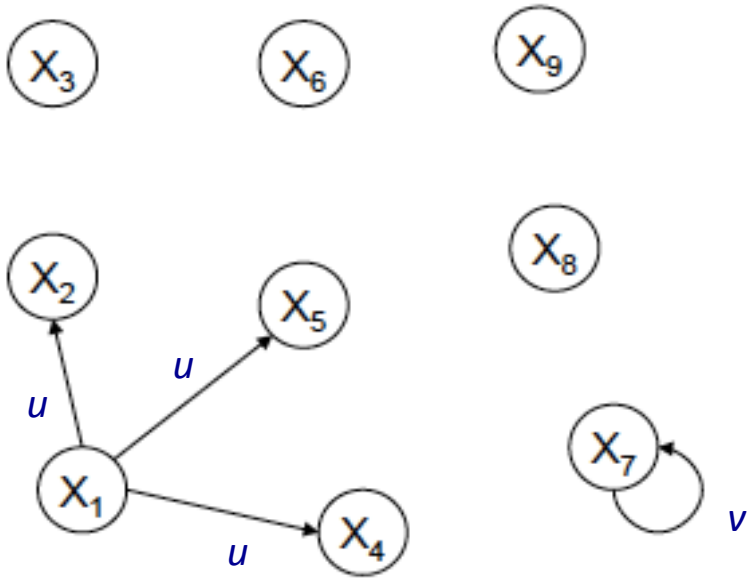
NB: Refinement of the partition now involves a scheme independent of *Pre*
(e.g., split the « bad » state into two)

[Tiwari-Khanna 2002]
[Habets-van Schuppen 2004][Belta-Habets 2006]
[Kloetzer-Belta HSCC 2006, TIMC 2012]

How to refine the quotient using *Post*

$Post_u(X)$ = { x' | $x' = f(x,u)$ for some x of X }

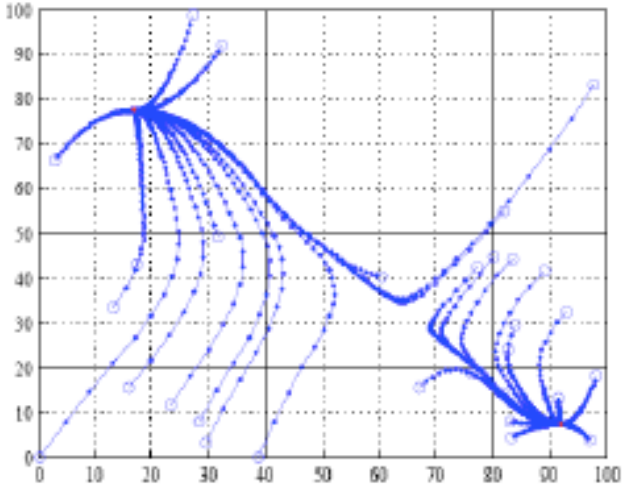




Transition $X - u \rightarrow X'$ is added to the quotient T'
when $Post_u(X) \cap X' \neq \emptyset$ in the original system T

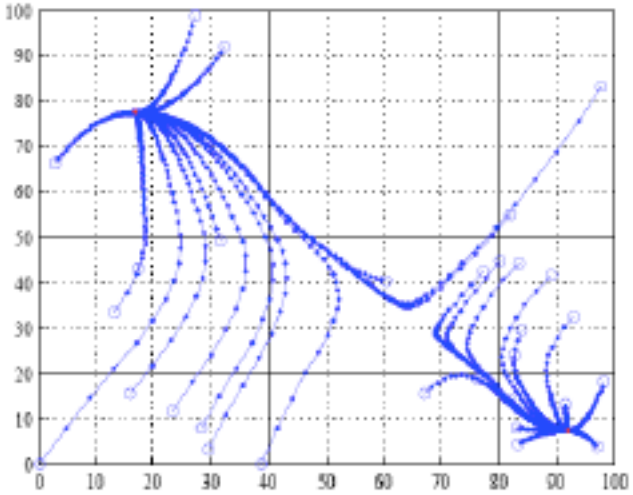
Verification for discrete-time PWA systems

Example: toggle switch



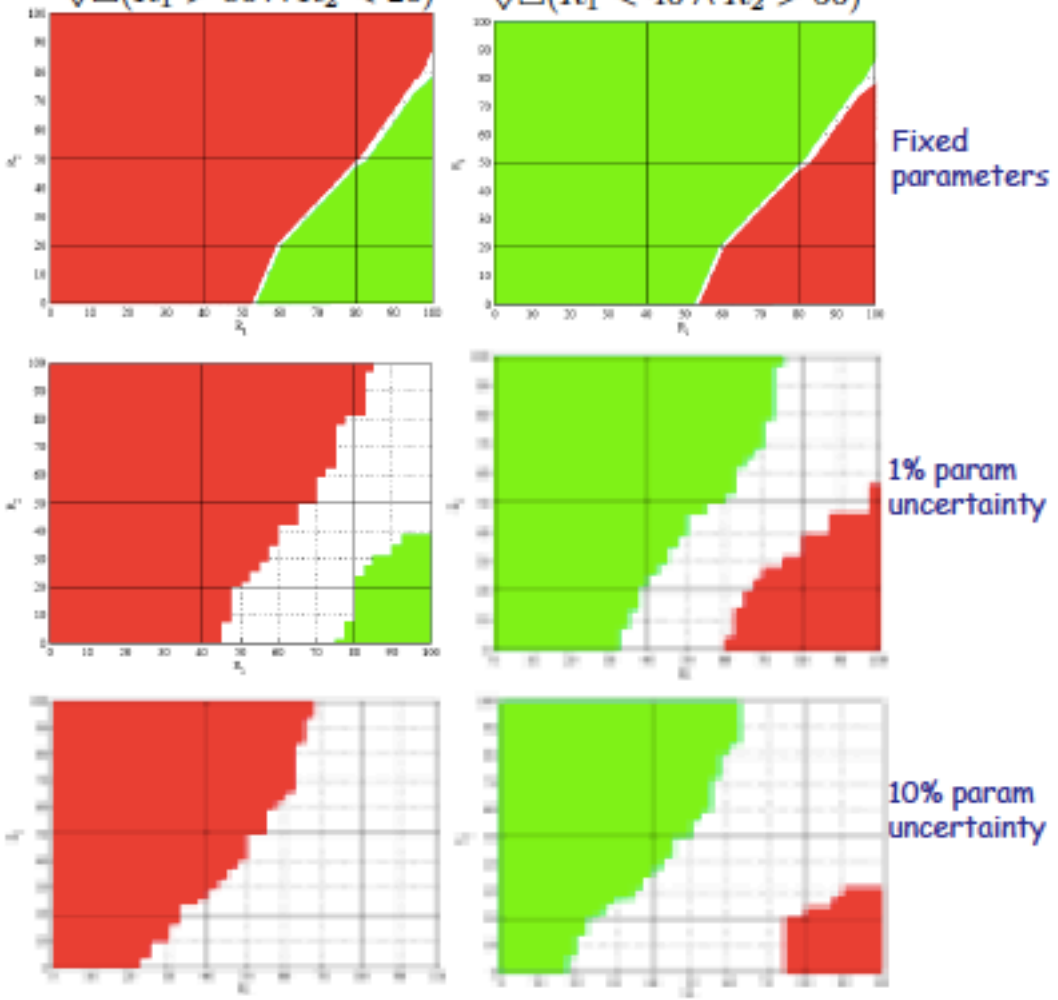
Verification for discrete-time PWA systems

Example: toggle switch



$\Diamond\Box(R_1 > 80 \wedge R_2 < 20)$

$\Diamond\Box(R_1 < 40 \wedge R_2 > 50)$



Fixed parameters

1% param uncertainty

10% param uncertainty

Initial satisfying states

Initial violating states

B. Yordanov and C. Belta, IEEE TAC 2010

Matlab tool: "FaPAS" (hyness.bu.edu/software)

V. MINIMATOR

https://bitbucket.org/ukuehne/minimator/wiki/Home

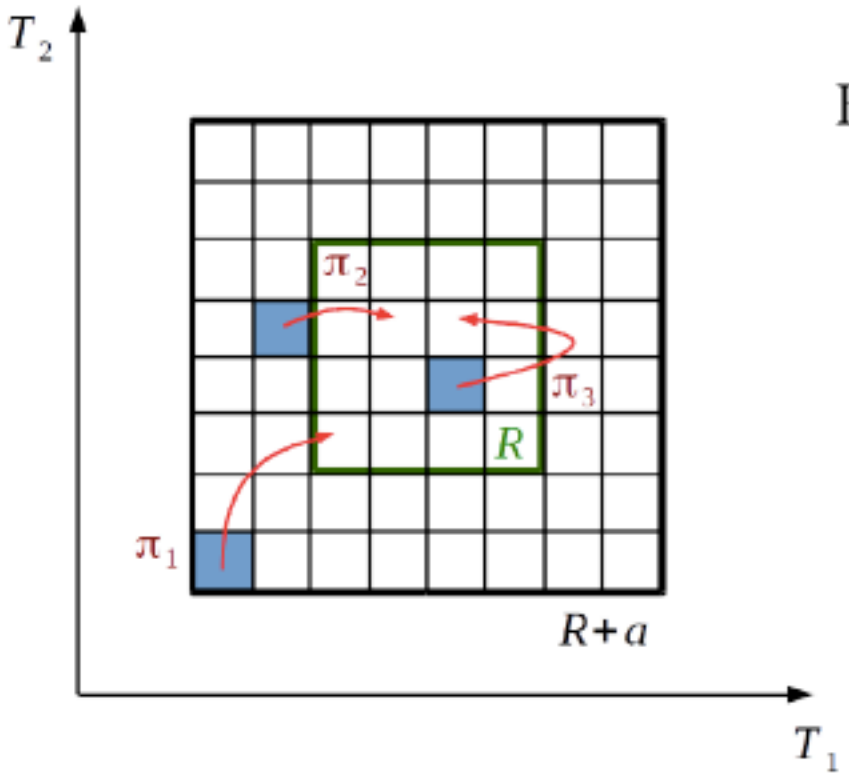


Romain Soulat, Ulrich Kühne, Adrien Le Coënt

Centralized control synthesis

$$x(t+1) = f(x(t), u)$$

Goal: from any $x \in R + a$, reach the target zone R.



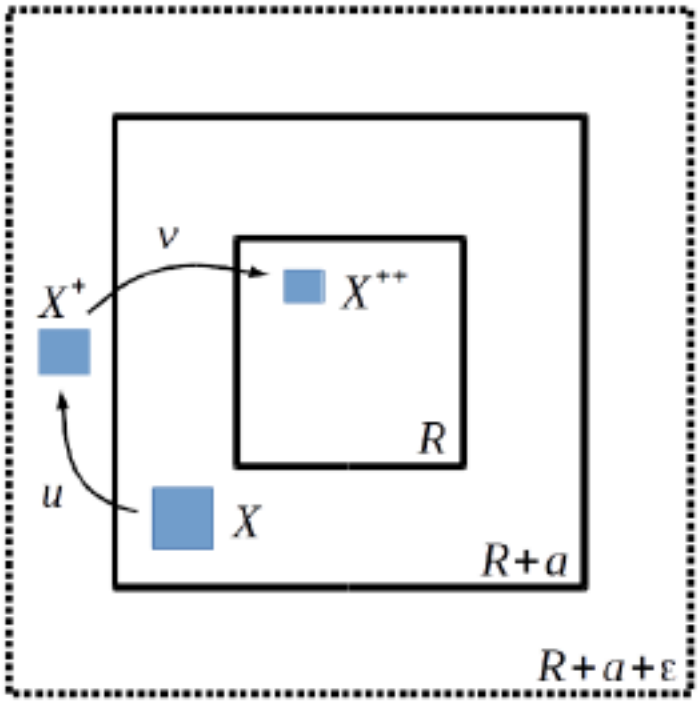
Basic idea:

- Generate a tiling of $R + a$
- Look for patterns (input sequences) mapping the tiles into R
- If it fails, generate another tiling.

Centralized control synthesis

$$x(t+1) = f(x(t), u)$$

Example of a validated pattern of length 2 mapping the tile X into R with a tolerance in $R + a + \varepsilon$:



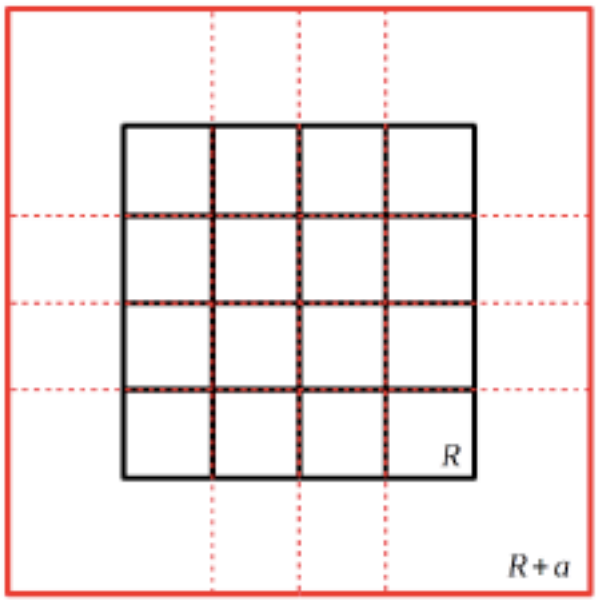
spec(X):

- $X \subset R + a$
- $X^+ = f(X, u) \subset R + a + \varepsilon$
- $X^{++} = f(X^+, v) \subset R$

- Pattern $u \cdot v$ depends only on X

Reachability

Parametric extension of a tiling:



Problem to solve: Find (the maximum value of) $a \geq 0$ such that $R + a$ can be mapped into R.

$\Rightarrow$ Can be solved by constrained optimization algorithms

Basic algorithm

target *tolerance* *time horizon* *initial zone*

Input: R ε K Output: a tiling P of $R+a$ satisfying *spec* (for some $a \geq 0$)

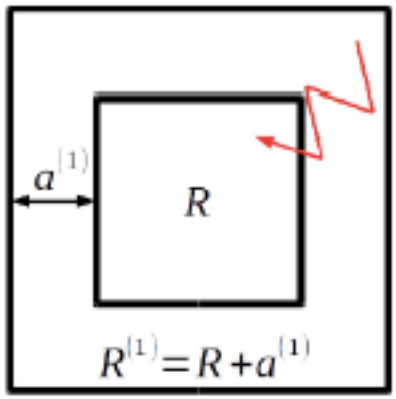
Initially, $P := [R+a]$
while *true*
    if some tile X of P violates $spec(X)$,
        refine P by splitting X endif
endwhile

If the algorithm terminates, we have $spec(X)$ for each tile X of P with:

$spec(X)$ = *reachability of* R *from* X *in* K *steps while always staying inside* $R+a+\varepsilon$
= $Post_\pi(X) \subset R$ for some pattern π of length $\leq K$
$\wedge$ $Post_{\pi'}(X) \subset R+a+\varepsilon$ for all prefix π' of π

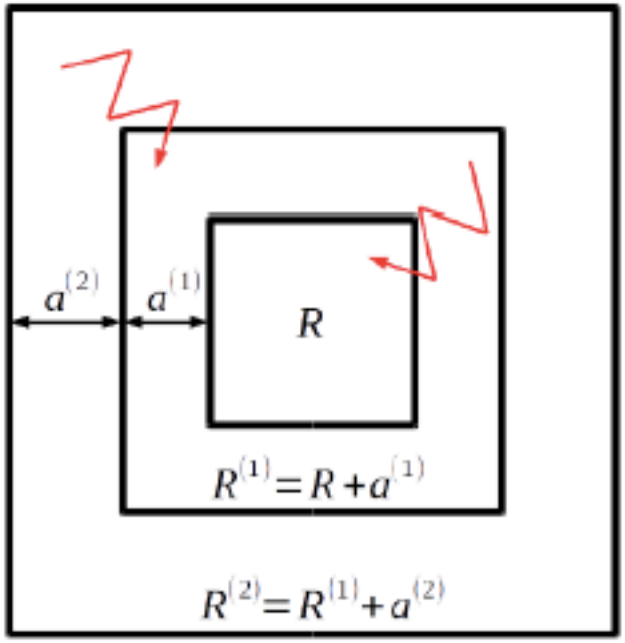
NB: if $a=0$, *spec* states the *invariance* of R (with tolerance ε)

Reachability: backward iteration of the procedure



Iterated control of $R^{(1)} = R + a^{(1)}$ towards R,

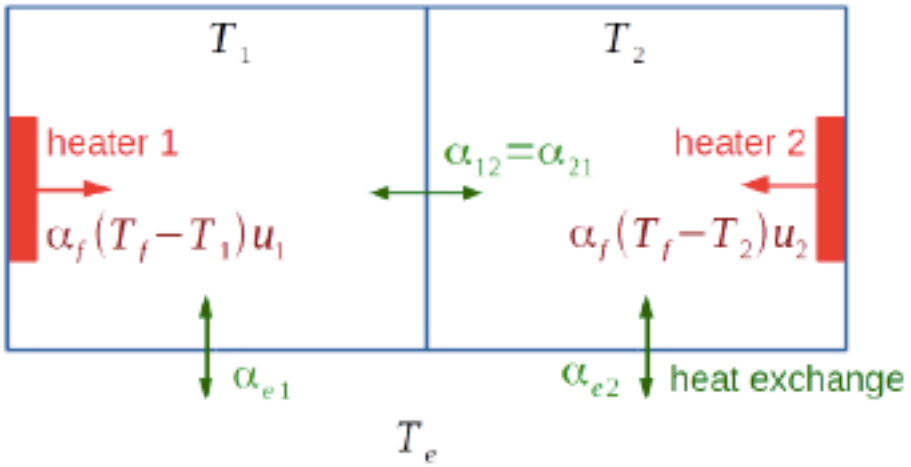
Reachability: backward iteration of the procedure



Iterated control of $R^{(1)} = R + a^{(1)}$ towards R, and $R^{(2)} = R^{(1)} + a^{(2)}$ towards $R^{(1)}$.

$\Rightarrow$ Compute a basin of attraction of R

Example: Two-room apartment



$$T_1(t+1) = f_1(T_1(t), T_2(t), u_1)$$
$$T_2(t+1) = f_2(T_1(t), T_2(t), u_2)$$

- 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.

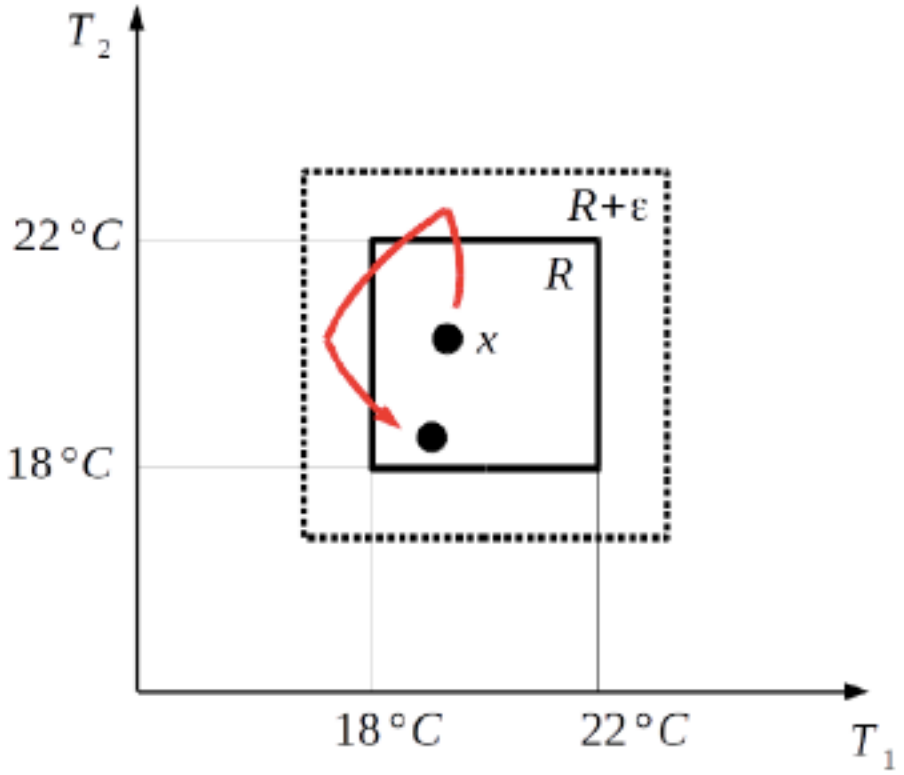
Reachability and Stability Properties for the two-room apartment

Input: R, ε

Output: a, controlled tiling of $R + a$

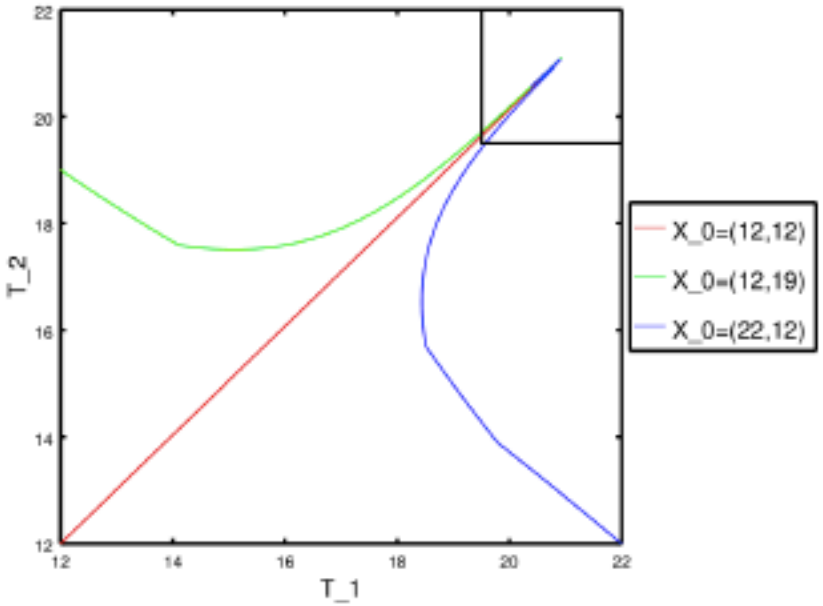
Guaranteed properties: reachability from $R + a$ to R, stability in $R + \varepsilon$, safety in $R + a + \varepsilon$

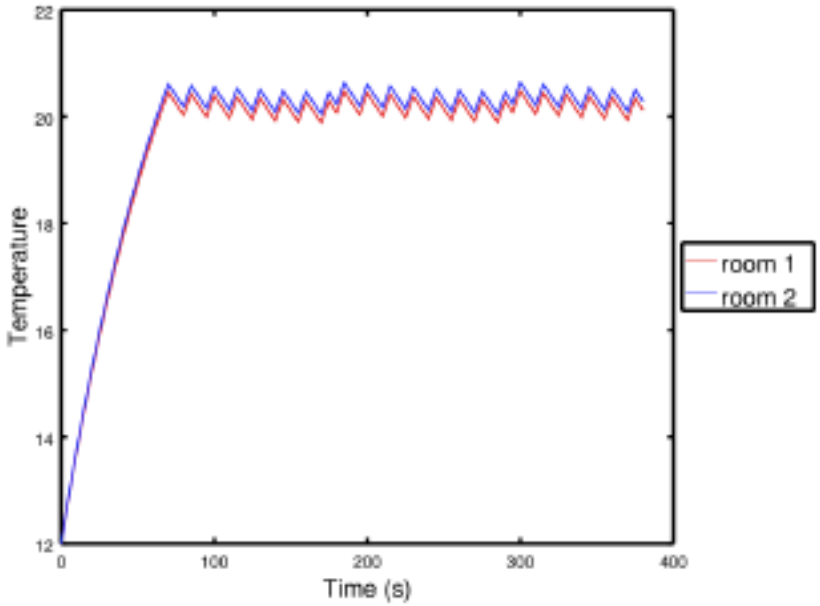
- Stability: special case of reachability, with $a = 0$.



Centralized control

Input: $R = [18.5, 22]^2$, $\varepsilon = 1.5$
Output: $a = 6$ in 4 steps, cpu time: $\sim$ 20s





Simulations of the centralized reachability controller for three different initial conditions plotted in the state space plane (left); simulation of the centralized reachability controller for the initial condition $(12, 12)$ plotted within time (right).

VI. Compositionality

Switched Systems

We suppose that the system can be written:

$$x_1(t+1) = f_1(x_1(t), x_2(t), u_1)$$
$$x_2(t+1) = f_2(x_1(t), x_2(t), u_2)$$

- First component of the state $x_1 \in \mathbb{R}^{n_1}$
- Second component of the state $x_2 \in \mathbb{R}^{n_2}$

$$n = n_1 + n_2$$

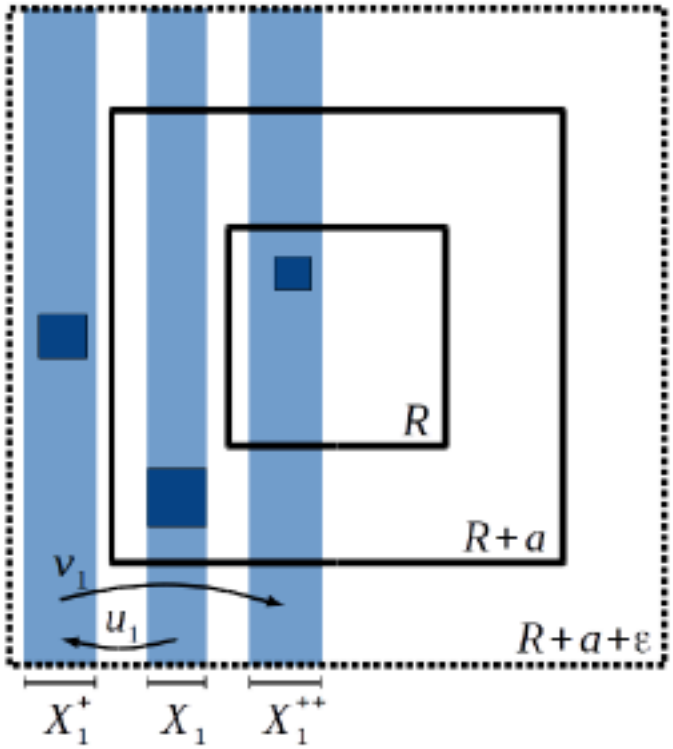
- First component of the control $u_1 \in U_1$ with $|U_1| = N_1$
- Second component of the control $u_2 \in U_2$ with $|U_2| = N_2$

$$U = U_1 \times U_2$$

Distributed control synthesis

$$x_1(t+1) = f_1(x_1(t), x_2(t), u_1)$$
$$x_2(t+1) = f_2(x_1(t), x_2(t), u_2)$$

Target zone: $R = R_1 \times R_2$



spec1(X1):

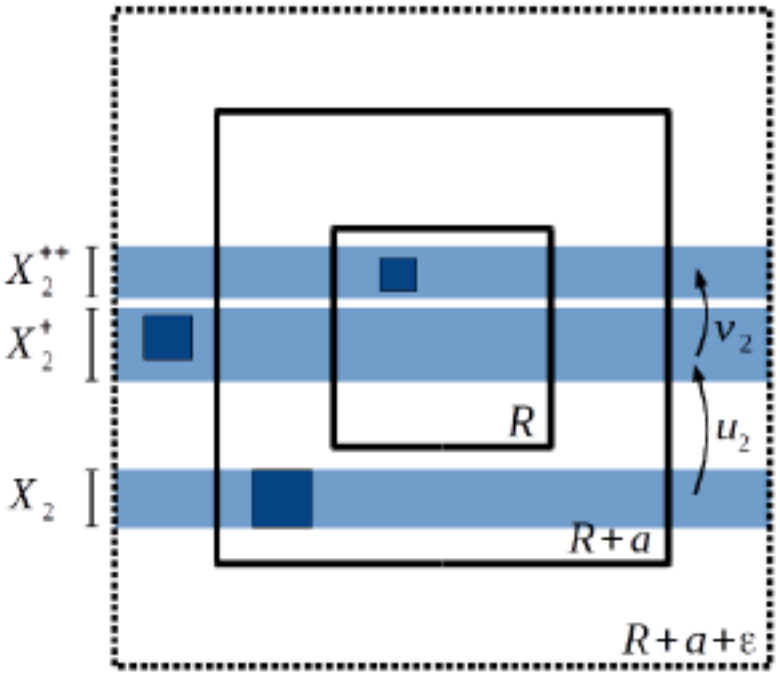
- $X_1 \subset R_1 + a$
- $X_1^+ = f_1(X_1, R_2 + a, u_1) \subset R_1 + a + \varepsilon$
- $X_1^{++} = f_1(X_1^+, R_2 + a + \varepsilon, v_1) \subset R_1$

- Pattern $u_1 \cdot v_1$ depends only on X_1

Distributed control synthesis

$$x_1(t+1) = f_1(x_1(t), x_2(t), u_1)$$
$$x_2(t+1) = f_2(x_1(t), x_2(t), u_2)$$

Target zone: $R = R_1 \times R_2$



spec2(X2):

- $X_2 \subset R_2 + a$
- $X_2^+ = f_2(R_1 + a, X_2, u_2) \in R_2 + a + \varepsilon$
- $X_2^{++} = f_2(R_1 + a + \varepsilon, X_2^+, v_2) \in R_2$

☐ Pattern $u_2 \cdot v_2$ depends only on X_2

Soundness of the distributed control synthesis

If *spec1(X1)* and *spec2(X2)* are *true*, ie.:

- $X_1 \subset R_1 + a$
- $X_1^+ = f_1(X_1, R_2 + a, u_1) \subset R_1 + a + \varepsilon$
- $X_1^{++} = f_1(X_1^+, R_2 + a + \varepsilon, v_1) \subset R_1$

and

- $X_2 \subset R_2 + a$
- $X_2^+ = f_2(R_1 + a, X_2, u_2) \in R_2 + a + \varepsilon$
- $X_2^{++} = f_2(R_1 + a + \varepsilon, X_2^+, v_2) \in R_2$

Then *spec(X)* (with *X=(X1,X2)*) is *true*, ie.:

- $X \subset R + a$
- $X^+ = f(X, u) \subset R + a + \varepsilon$
- $X^{++} = f(X^+, v) \subset R$

with *u=(u1,u2)* and *v=(v1,v2)*

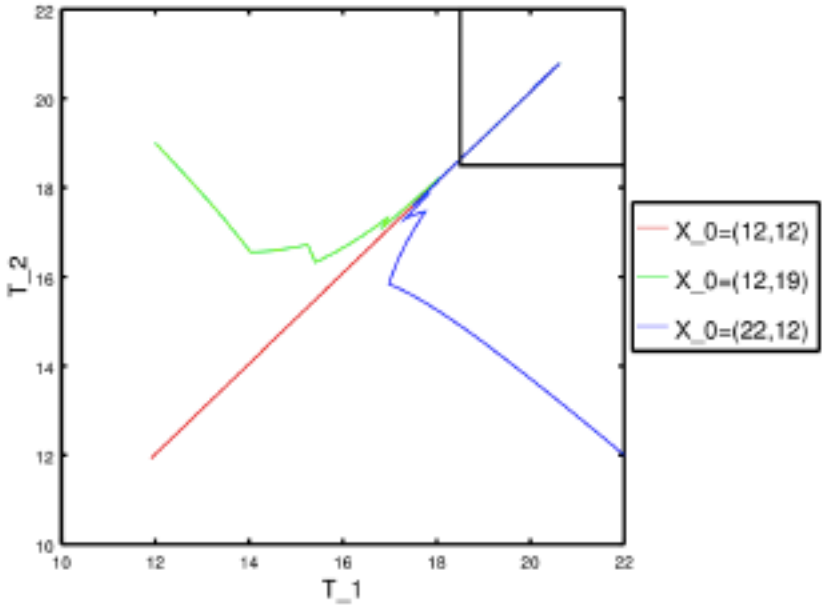
Hence the distributed control achieves the goal of the centralized control

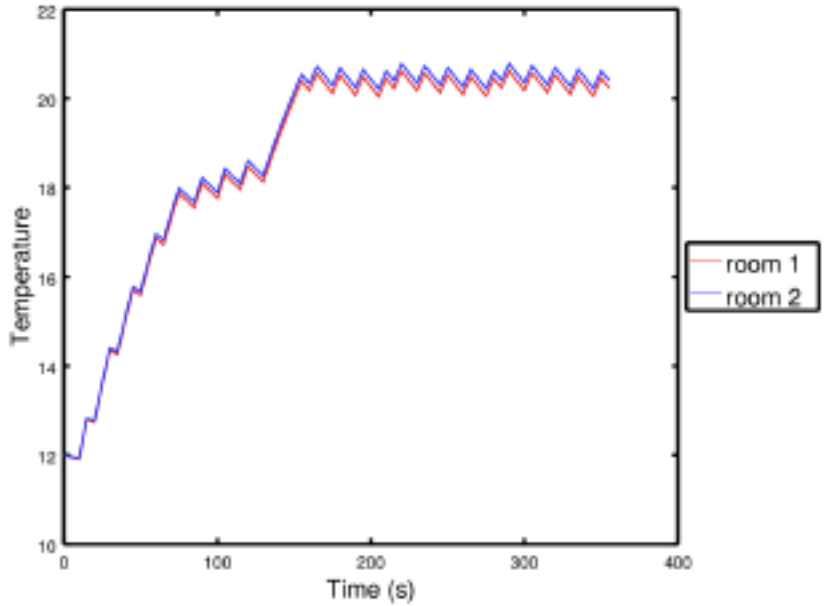
Advantage: the state dimension *n* and the nb of modes *N* have split in 2

Condition: requires the *weak interdependency* of the arguments of *f*

Distributed control

Input: $R = [18.5, 22]^2$, $\varepsilon = 1.5$
Output: $a = 6$ in 4 steps, cpu time: $\sim$ 20s





Simulations of the distributed reachability controller for three different initial conditions plotted in the state space plane (left); simulation of the distributed reachability controller for the initial condition $(12, 12)$ plotted within time (right).

Seluxit case study

Kim G. Larsen, Marius Mikučionis, Marco Muniz, Jiri Srba, Jakob H. Taankvist. *Online and Compositional Learning of Controllers with Application to Floor Heating.* Tools and Algorithms for Construction and Analysis of Systems 2016.

Seluxit case study

Kim G. Larsen, Marius Mikučionis, Marco Muniz, Jiri Srba, Jakob H. Taankvist. *Online and Compositional Learning of Controllers with Application to Floor Heating.* Tools and Algorithms for Construction and Analysis of Systems 2016.

System dynamics:

$$\frac{d}{dt}T_i(t) = \sum_{j=1}^{n} A^d_{i,j}(T_j(t) - T_i(t)) + B_i(T_{env}(t) - T_i(t)) + H_{i,j}.v_j$$

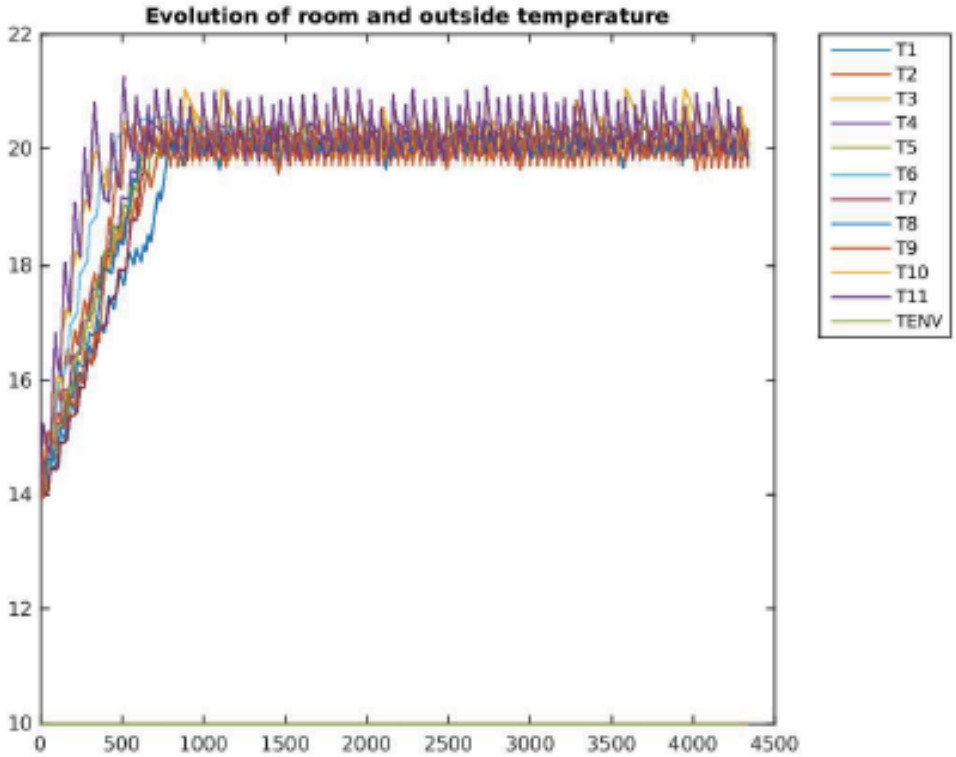
- System of dimension 11
- 2^{11} combinations of v_j (not all admissible, constraint on the number of open valves)
- Pipes heating a room may influence other rooms
- Doors opening and closing (here: average between open and closed)
- Varying external temperature (here: $T_{env} = 10°C$)
- Measures and switching every 15 minutes

Seluxit case study, guaranteed reachability and stability

Decomposition in 5 + 6 rooms (cf. [Larsen et al., TACAS 2016], thanks to the Aalborg team for the simulator)

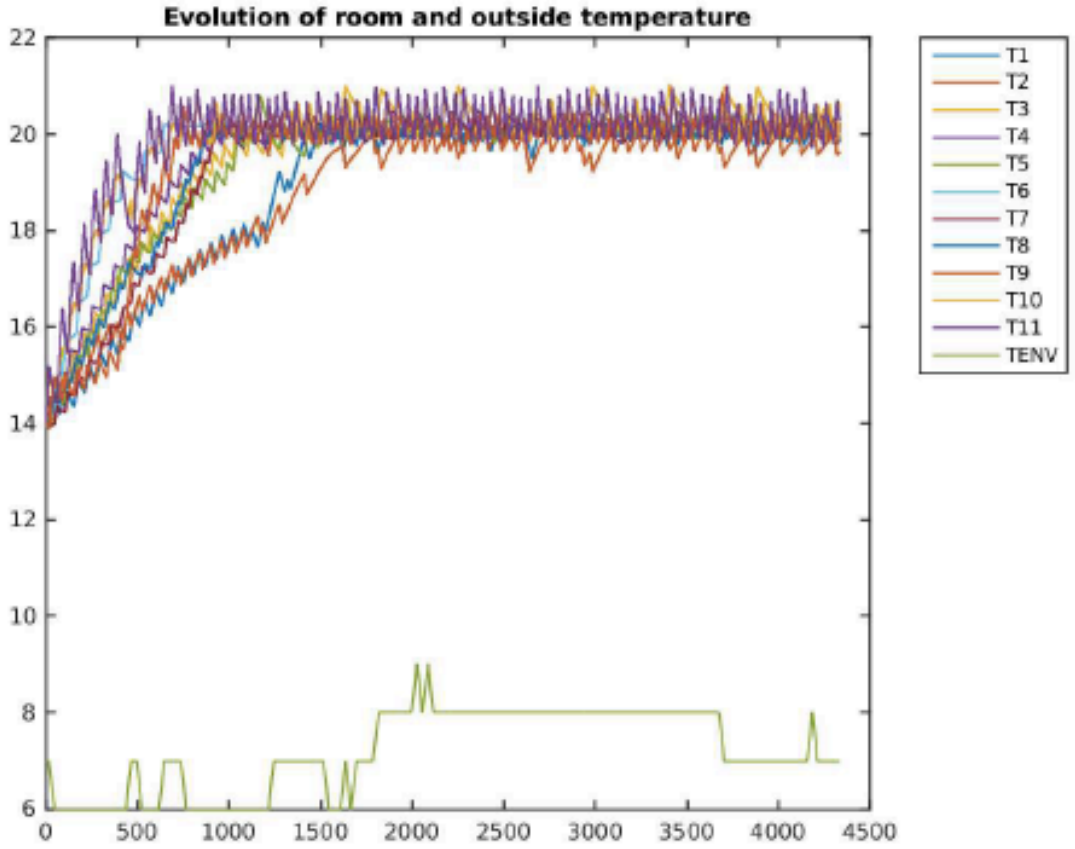
Input:
$R = [18, 22]^{11}$
$\varepsilon = 0.5$
$T_{env} = 10$

Output:
$a = 4$ in 15 steps
cpu time: 6h



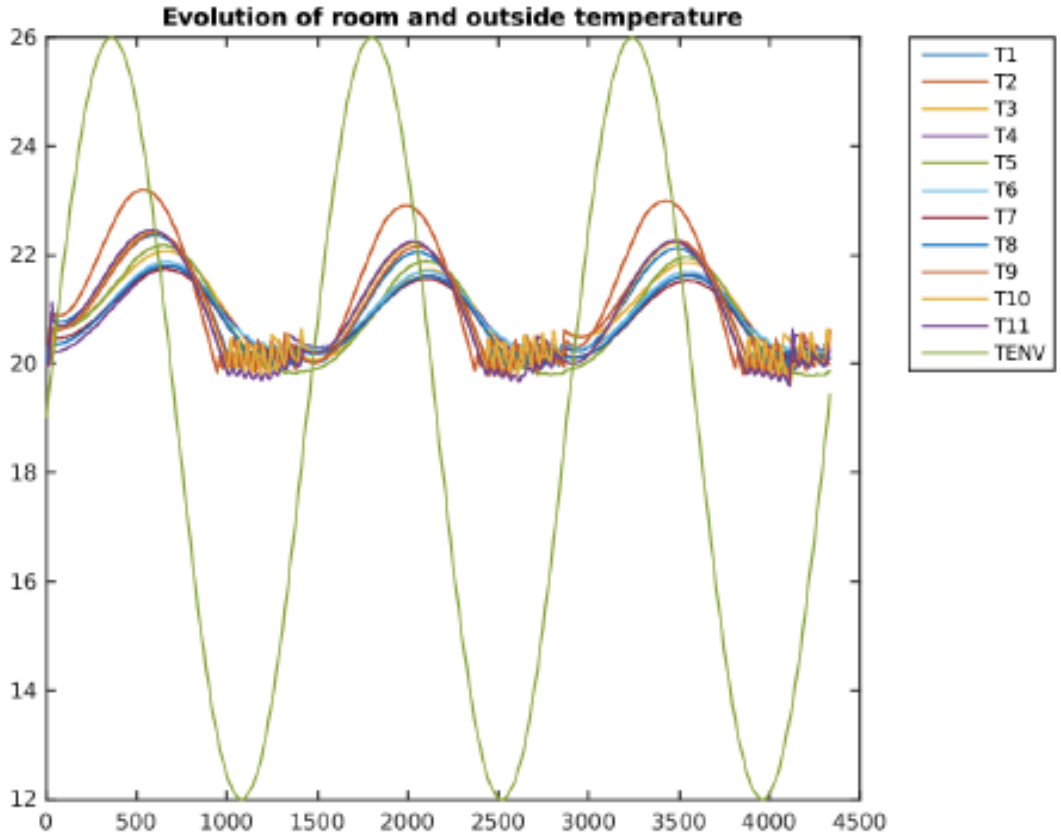
Simulation of the Seluxit case study plotted with time (in min) for $T_{env} = 10°C$.

Seluxit case study, robustness test



Simulation of the Seluxit case study in the soft winter scenario.

Seluxit case study, robustness test (2)



Simulation of the Seluxit case study in the spring scenario.

VII. Model reduction

Model order reduction

Original system :

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Construction of a reduced order system $\hat{\Sigma}$ of lower dimension :

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t), \qquad (1)$$

Reduction by Balanced Truncation [Antoulas, Gucercin, 2004] : $\hat{x} = \pi_r x$

Synthesis of the control rule $u(\cdot)$ at the low-order level and application at the full-order level.

Requirements :

- bounding of the error $\varepsilon_r = |Post_{Pat}(\hat{x}) - \pi_r Post_{Pat}(x)|$

Reduced order control synthesis

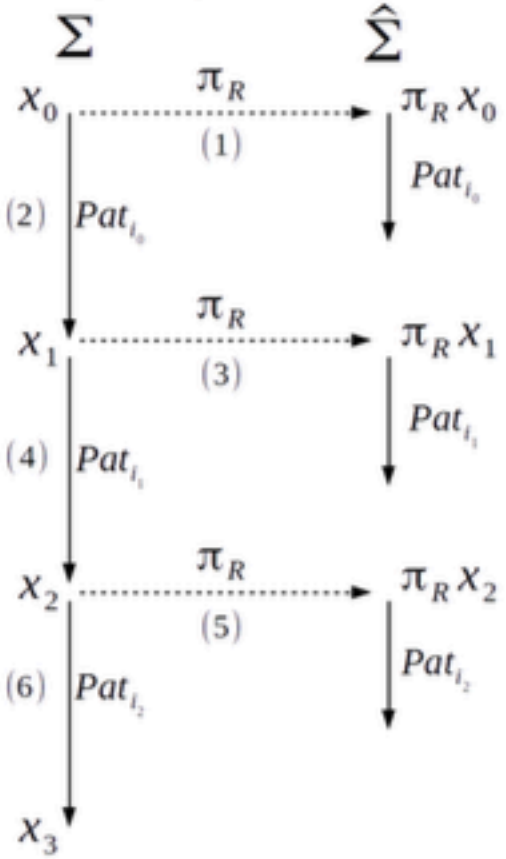
Input : $\hat{R}, \varepsilon_r$ Output : a tiling $\hat{P}$ of $\hat{R}$ satisfying $spec(X)$

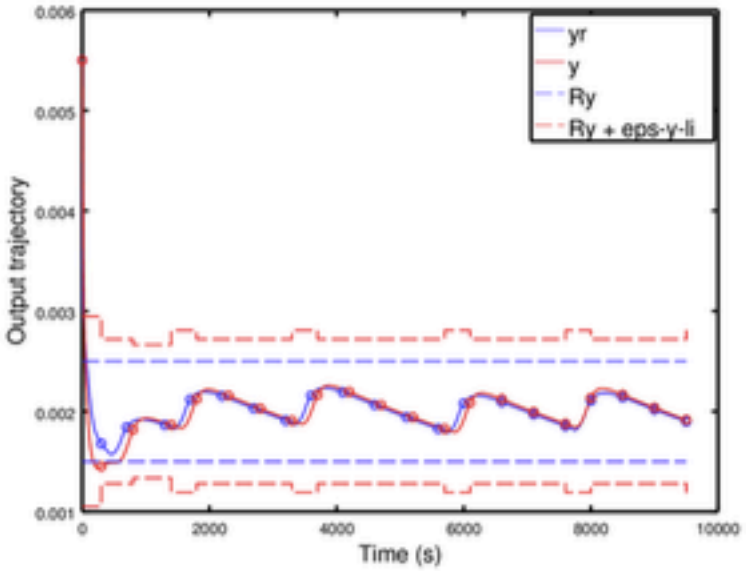
Initially, $P := R$
while *true*
  if some state $\hat{X}$ of $\hat{P}$ violates $spec(X)$
  refine P by splitting $\hat{X}$
  endif
endwhile

$spec(X)$ takes into account the reduction error
Available for stability and attainability
ex (stability) : $spec(X) = Post_{Pat}(\hat{X}) \subseteq \hat{R} - \varepsilon_r$

Guaranteed on-line control

Simulation on a linearized model of a distillation column [Tong-Zhou-Wang-Mou14] : $n = 11$ and $n_r = 2$:





VIII. Many important issues not mentioned!

- Guards of hybrid systems (PWA)
- Non linearity
- Continuous-time dynamics
- Data structures (eg, zonotopes [Girard 2005])
- Observability
- Robustness
- Uncertainty
- Stochasticity

IX. Recapitulation

- Affine switched systems (special class of hybrid systems)
- Set-based approach
- Symbolic simulation using *Post*
- Compositionality

→ Safety-provable design with increasing scalability

(nb of continuous variables: $n \approx 3$ in 2004, $n \approx 5$ in 2010, $n \approx 11$ in 2016, …)

References

- D. Bertsekas, I. Rhodes. On the minimax reachability of target sets and target tubes. Automatica, 1971

- A Bouajjani, JC Fernandez, N Halbwachs. Minimal model generation, CAV 1990

- I. Mitchell, A. Bayen, C. Tomlin. A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games, IEEE TAC 2004

- B. Yordanov, X. Belta. Formal analysis of discrete-time piecewise affine systems. IEEE TAC 2010

- A. Le Coënt, L. Fribourg, N. Markey, F. De Vuyst, L. Chamoin. Distributed Synthesis of State-Dependent Switching Control. RP'16, LNCS 9899, Springer, 2016

- A. Le Coënt, F. De Vuyst, C. Rey, L. Chamoin, L. Fribourg. Control of mechanical systems using set-based methods. Int. J. of Dynamics and Control, 2016