Guaranteed control synthesis for switched space-time dynamical systems

Soutenance de thèse de doctorat en mathématiques appliquées,

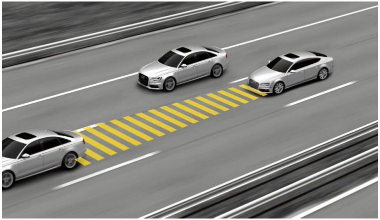
CMLA, ENS Paris-Saclay

Adrien Le Coënt

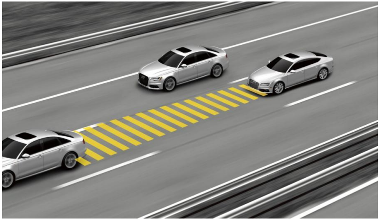
Sous la supervision de Florian De Vuyst, Ludovic Chamoin, Laurent Fribourg, Christian Rey

July 10, 2018

Context: control systems



Context: control systems



Issues: guaranteed control synthesis for safety/stability/reachability of safety critical systems, nonlinear systems, PDEs

Outline

1 **Guaranteed control of switched systems**
- Switched systems
- Control of switched systems
- State-space bisection algorithm

2 The reachable set computation

3 Distributed synthesis of controllers

4 Control of partial differential equations with model order reduction

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

- state $x \in \mathbb{R}^n$

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

- state $x \in \mathbb{R}^n$
- bounded perturbation $d(\cdot) : \mathbb{R}^+ \longrightarrow D \subset \mathbb{R}^m$

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

- state $x \in \mathbb{R}^n$
- bounded perturbation $d(\cdot) : \mathbb{R}^+ \longrightarrow D \subset \mathbb{R}^m$
- switching signal $\sigma(\cdot) : \mathbb{R}^+ \longrightarrow U$ (piecewise constant)

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

- state $x \in \mathbb{R}^n$
- bounded perturbation $d(\cdot) : \mathbb{R}^+ \longrightarrow D \subset \mathbb{R}^m$
- switching signal $\sigma(\cdot) : \mathbb{R}^+ \longrightarrow U$ (piecewise constant)
- $U = \{1, \ldots, N\}$ finite set of modes, associated with the dynamics

$$\dot{x}(t) = f_u(x(t), d(t)), \; u \in U$$

Switched Systems

A continuous-time switched system

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

is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

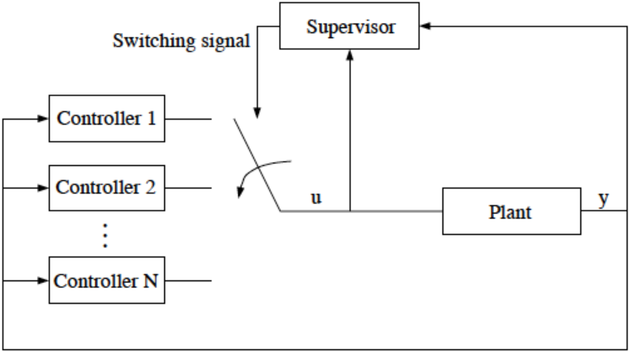
- state $x \in \mathbb{R}^n$
- bounded perturbation $d(\cdot) : \mathbb{R}^+ \longrightarrow D \subset \mathbb{R}^m$
- switching signal $\sigma(\cdot) : \mathbb{R}^+ \longrightarrow U$ (piecewise constant)
- $U = \{1, \dots, N\}$ finite set of modes, associated with the dynamics

$$\dot{x}(t) = f_u(x(t), d(t)), \ u \in U$$

We focus on sampled switched systems: switching instants occur periodically every τ, i.e. σ is constant on $[i\tau, (i+1)\tau)$

Examples of switched systems

Controlled Switched Systems: Schematic View



Control of switched systems: possible approaches

- Optimal control [Trélat, Rubio, Fattorini,...]
  - Minimization under constraints of a cost function

Control of switched systems: possible approaches

- Optimal control [Trélat, Rubio, Fattorini,...]
  - Minimization under constraints of a cost function
- Lyapunov approaches [Krstic, Teel, Coron, Sigalotti, Liberzon...]
  - Stability analysis and stabilization using Lyapunov functions

Control of switched systems: possible approaches

- Optimal control [Trélat, Rubio, Fattorini,...]
  - Minimization under constraints of a cost function
- Lyapunov approaches [Krstic, Teel, Coron, Sigalotti, Liberzon...]
  - Stability analysis and stabilization using Lyapunov functions
- Symbolic methods

Control of switched systems: possible approaches

- Optimal control [Trélat, Rubio, Fattorini,...]
  - Minimization under constraints of a cost function
- Lyapunov approaches [Krstic, Teel, Coron, Sigalotti, Liberzon...]
  - Stability analysis and stabilization using Lyapunov functions
- Symbolic methods
  - Finite state abstraction methods:
    - Approximate bisimulation (PESSOA/CoSyMA) [Tabuada,Girard...]
    - Feedback refinement relations (SCOTS) [Zamani, Rungger, Reissig...]

Control of switched systems: possible approaches

- Optimal control [Trélat, Rubio, Fattorini,...]
  - Minimization under constraints of a cost function
- Lyapunov approaches [Krstic, Teel, Coron, Sigalotti, Liberzon...]
  - Stability analysis and stabilization using Lyapunov functions
- Symbolic methods
  - Finite state abstraction methods:
    - Approximate bisimulation (PESSOA/CoSyMA) [Tabuada,Girard...]
    - Feedback refinement relations (SCOTS) [Zamani, Rungger, Reissig...]
  - Tiling methods:
    - Uniform tiling [Girard, Witrant, Arcak...]
    - Adaptive state-space bisection (MINIMATOR) [Soulat, Fribourg]

Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

At each sampling time $k\tau$, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

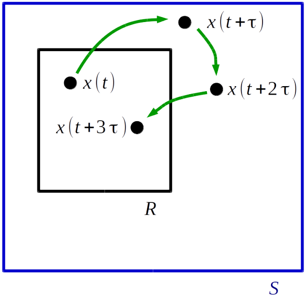
Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

At each sampling time $k\tau$, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

Given two sets R, S:

- (R, S)-stability: $x(t)$ returns in R infinitely often, at some multiples of sampling period τ, and always stays in S



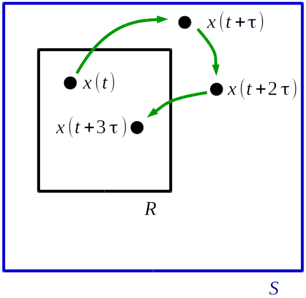
Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

At each sampling time $k\tau$, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

Given two sets R, S:

- (R, S)-stability: $x(t)$ returns in R infinitely often, at some multiples of sampling period τ, and always stays in S



NB: classic stabilization impossible here (no common equilibrium pt)
$\rightsquigarrow$ practical stability

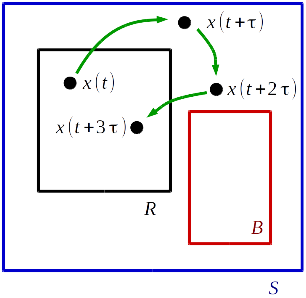
Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

At each sampling time $k\tau$, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

Given three sets R, B, S:

- (R, B, S)-avoidance: $x(t)$ returns in R infinitely often, at some multiples of sampling period τ, and always stays in $S \backslash B$



NB: classic stabilization impossible here (no common equilibrium pt)
$\rightsquigarrow$ practical stability

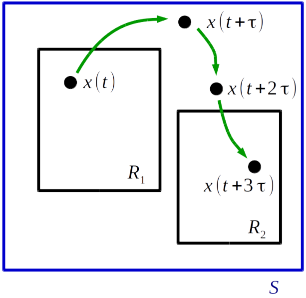
Control Synthesis Problem

We consider the state-dependent control problem of synthesizing σ:

At each sampling time $k\tau$, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

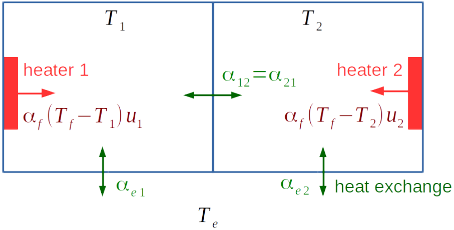
Given three sets R_1, R_2, S:

- (R_1, R_2, S)-reachability: $x(t)$ starting in R_1 reaches R_2 after some multiples of sampling period τ, and always stays in S



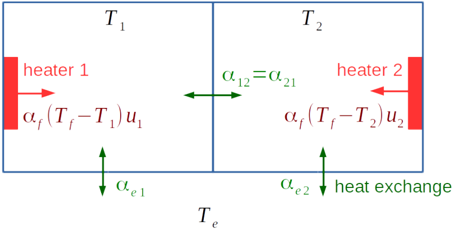
NB: classic stabilization impossible here (no common equilibrium pt)
$\rightsquigarrow$ practical stability

Example: Two-room apartment



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

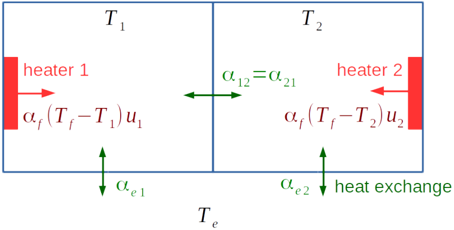
Example: Two-room apartment



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

- 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

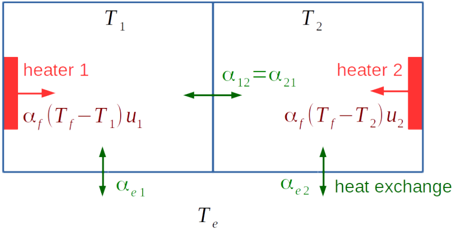
Example: Two-room apartment



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

- 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



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

- 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.

(R, S)-stability property for the two-room apartment

Input:

- R, S
- an integer K (maximal length of patterns)

Output: controlled covering of R (each covering set is coupled with a pattern)

Guaranteed properties: (R, S)-stability

(R, S)-stability property for the two-room apartment

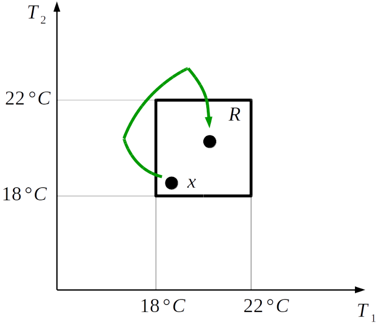
Input:

- R, S
- an integer K (maximal length of patterns)

Output: controlled covering of R (each covering set is coupled with a pattern)

Guaranteed properties: (R, S)-stability

- Recurrence in R: after some ($\leq K$) steps of time, the temperature returns in R



(R, S)-stability property for the two-room apartment

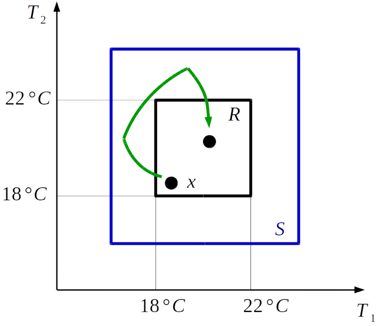
Input:

- R, S
- an integer K (maximal length of patterns)

Output: controlled covering of R (each covering set is coupled with a pattern)

Guaranteed properties: (R, S)-stability

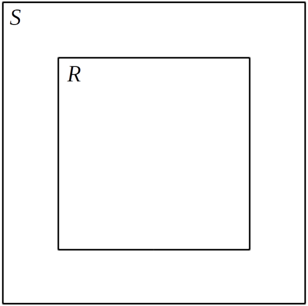
- Recurrence in R: after some ($\leq K$) steps of time, the temperature returns in R
- Safety in S: $x(t)$ always stays in S.



Control tiling procedure

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

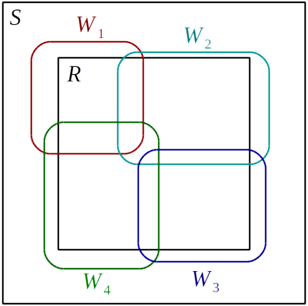
Goal: from any $x \in R$, return in R while always staying in S.



Control tiling procedure

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

Goal: from any $x \in R$, return in R while always staying in S.



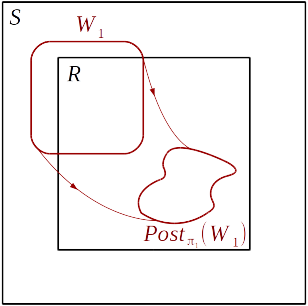
Basic idea:

- Generate a covering of R

Control tiling procedure

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

Goal: from any $x \in R$, return in R while always staying in S.



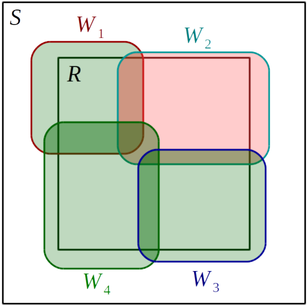
Basic idea:

- Generate a covering of R
- Look for patterns (input sequences) mapping the tiles into R while always staying in S

Control tiling procedure

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

Goal: from any $x \in R$, return in R while always staying in S.



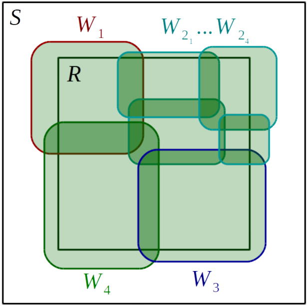
Basic idea:

- Generate a covering of R
- Look for patterns (input sequences) mapping the tiles into R while always staying in S
- If it fails,

Control tiling procedure

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

Goal: from any $x \in R$, return in R while always staying in S.



Basic idea:

- Generate a covering of R
- Look for patterns (input sequences) mapping the tiles into R while always staying in S
- If it fails, generate another covering.

Limits

- Requires the computation of the reachable set
  - unknown in general
  - can be approximated using numerical schemes and/or strong hypotheses

Limits

- Requires the computation of the reachable set
  - unknown in general
  - can be approximated using numerical schemes and/or strong hypotheses
- High computational complexity (curse of dimensionality):
  - m covering sets, patterns of length K, N switched modes $\Rightarrow$ cost in $O(mN^K)$
  - using a bisection heuristics of depth D in dimension n $\Rightarrow$ cost in $O(2^{nD}N^K)$

Limits

- Requires the computation of the reachable set
  - unknown in general
  - can be approximated using numerical schemes and/or strong hypotheses
- High computational complexity (curse of dimensionality):
  - m covering sets, patterns of length K, N switched modes $\Rightarrow$ cost in $O(mN^K)$
  - using a bisection heuristics of depth D in dimension n $\Rightarrow$ cost in $O(2^{nD}N^K)$

We propose:

- Handling nonlinear dynamics without strong hypotheses with guaranteed numerical schemes
- Handling higher dimensions using compositionality
- Synthesizing controllers for PDEs using Model Order Reduction

Outline

1 Guaranteed control of switched systems

2 The reachable set computation

3 Distributed synthesis of controllers

4 Control of partial differential equations with model order reduction

Outline

1 Guaranteed control of switched systems

2 The reachable set computation
- State-of-the-art and validated simulation
- Euler approximate solutions
- Case studies

3 Distributed synthesis of controllers

4 Control of partial differential equations with model order reduction

Numerical integration, reachability analysis

- Classical (non guaranteed) methods: Euler, Runge-Kutta, implicit, explicit schemes...
- Guaranteed reachability analysis: Enclosing solutions, error bounding, additional hypotheses

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:
  - Monotonicity, ISS, incremental stability [Girard, Sontag, Zamani, Tabuada...]
  - Validated simulation, guaranteed integration [Moore, Lohner, Bertz, Makino, Nedialkov, Jackson, Corliss, Chen, Ábrahám, Sankaranarayanan, Taha, Chapoutot,...]
  - Sensitivity Analysis [Donzé, Maler...]

Numerical integration, reachability analysis

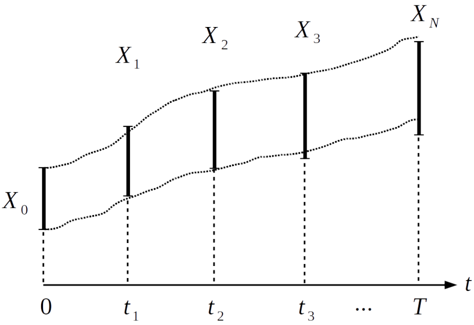
- Classical (non guaranteed) methods: Euler, Runge-Kutta, implicit, explicit schemes...
- Guaranteed reachability analysis: Enclosing solutions, error bounding, additional hypotheses
- State-of-the-art:
  - Monotonicity, ISS, incremental stability [Girard, Sontag, Zamani, Tabuada...]
  - Validated simulation,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

Validated simulation

DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$



Validated simulation

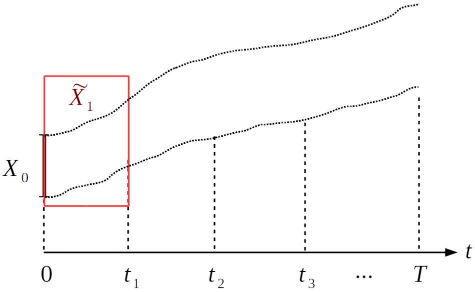
DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$



Validated simulation

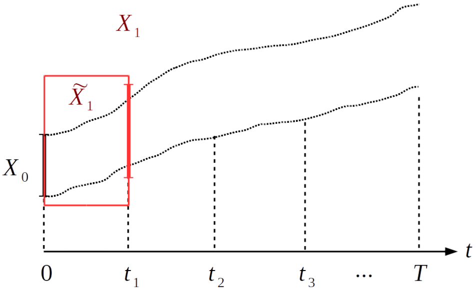
DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$
- Tighten the error $\|x_n - x(t_n; x_{n-1})\|$



Validated simulation

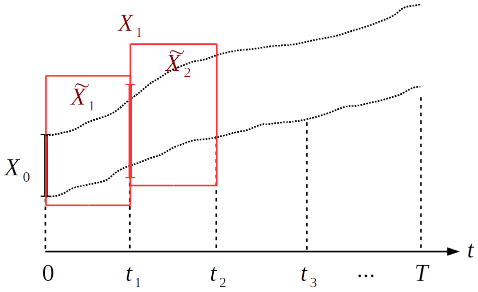
DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$
- Tighten the error $\|x_n - x(t_n; x_{n-1})\|$



Validated simulation

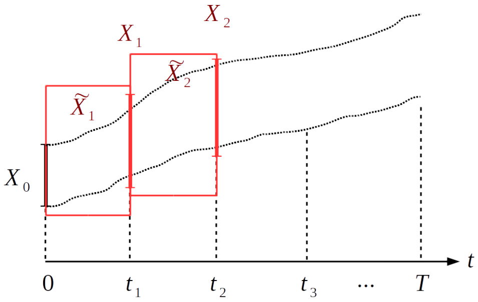
DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$
- Tighten the error $\|x_n - x(t_n; x_{n-1})\|$



Validated simulation

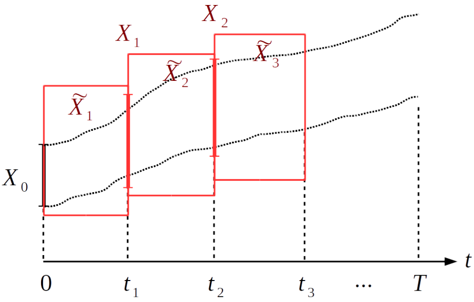
DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$
- Tighten the error $\|x_n - x(t_n; x_{n-1})\|$



Validated simulation

DynIBEX [Chapoutot, Alexandre dit Sandretto, 2016]

Runge-Kutta numerical scheme:

- Computation of a sequence of approximations (t_n, X_n) of the solution $X(t; X_0)$
- X_i computed with the previous step: $X_i = h(t_{i-1}, X_{i-1})$

Making it guaranteed:

- Enclose solutions (using Picard-Linedelöf operator and Banach's theorem) on $[t_{n-1}, t_n]$
- Tighten the error $\|x_n - x(t_n; x_{n-1})\|$

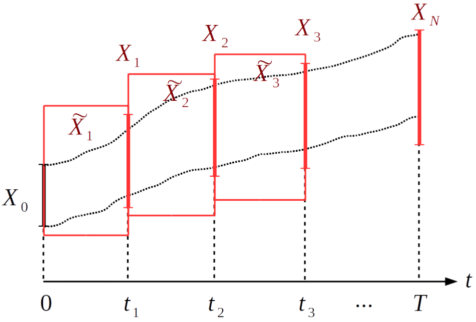
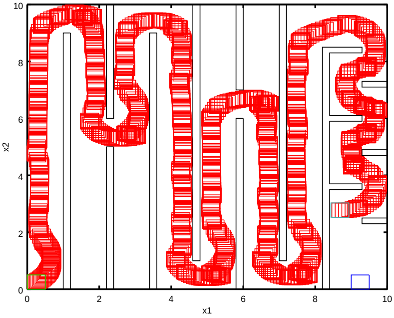
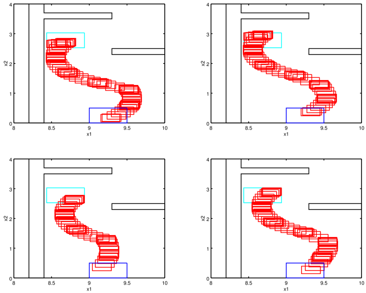


Illustration: a path planning problem

[Aström, Murray, 2010]

$$\dot{x} = v_0 \frac{\cos(\alpha+\theta)}{\cos(\alpha)}$$
$$\dot{y} = v_0 \frac{\sin(\alpha+\theta)}{\cos(\alpha)}$$
$$\dot{\theta} = \frac{v_0}{b}\tan(\delta)$$





Renewing the Euler scheme with the OSL property

$(H0)$ (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

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

NB: constants computed by constrained optimization.

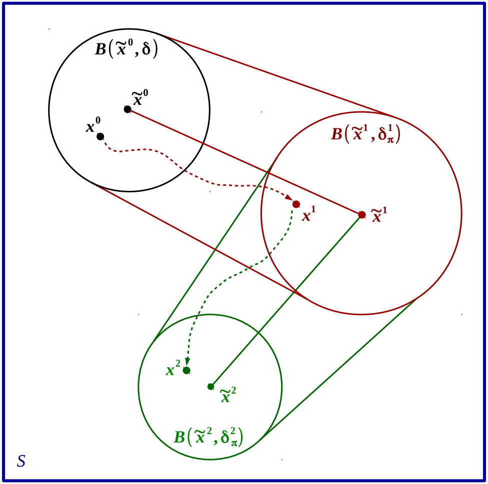
Main result

Theorem

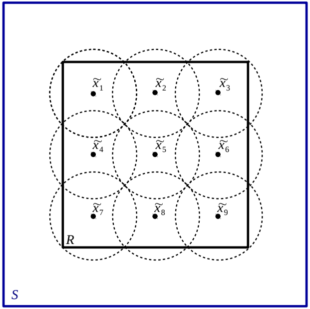
Given a sampled switched system satisfying (H0-H1), consider a point $\tilde{x}^0$ *and a positive real* δ. *We have, for all* $x^0 \in B(\tilde{x}^0, \delta)$, $t \in [0, \tau]$ *and* $j \in U$: $\phi_j(t; x^0) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t))$.
with

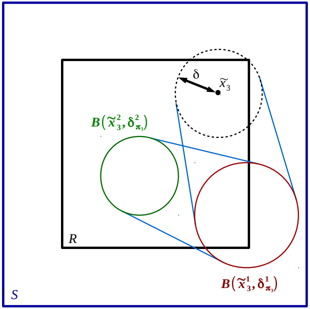
- *if* $\lambda_j < 0$: $\delta_j(t) = \left(\delta^2 e^{\lambda_j t} + \frac{C_j^2}{\lambda_j^2}\left(t^2 + \frac{2t}{\lambda_j} + \frac{2}{\lambda_j^2}\left(1 - e^{\lambda_j t}\right)\right)\right)^{\frac{1}{2}}$
- *if* $\lambda_j = 0$: $\delta_j(t) = \left(\delta^2 e^t + C_j^2(-t^2 - 2t + 2(e^t -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

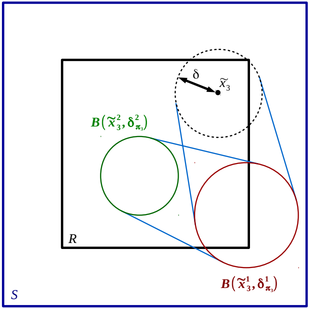


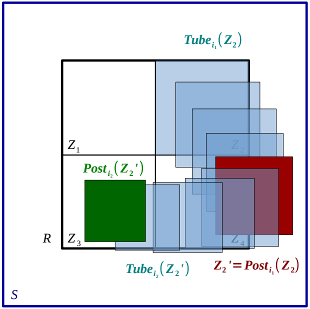
Control synthesis





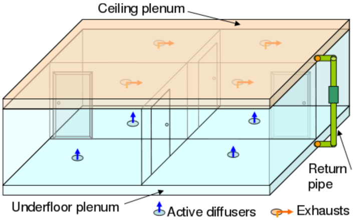
Euler vs Validated simulation





Building ventilation

[Meyer, Nazarpour, Girard, Witrant, 2014]



Four-room apartment, nonlinear dynamics:

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

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

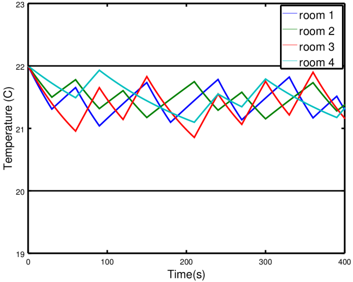
$\Rightarrow$ 16 switching modes

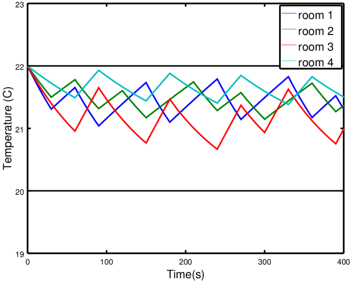
Building ventilation

	Euler	DynIBEX
R S	$[20, 22]^4$ $[19, 23]^4$	
τ	30	
Complete control	Yes	Yes
Number of balls/tiles Pattern length	4096 1	252 1
CPU time	63 seconds	249 seconds

Building ventilation

Simulation Euler vs validated simulation





Outline

1 Guaranteed control of switched systems

2 The reachable set computation

3 Distributed synthesis of controllers
- Distributed synthesis using zonotopes
- Distributed synthesis using Euler's method

4 Control of partial differential equations with model order reduction

Distributed control synthesis

Splitting of the system

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u_1) \\ \dot{x}_2 &= f_2(x_1, x_2, u_2) \end{aligned}$$

Control objective: (R, S)-stability with $R = R_1 \times R_2$, $S = S_1 \times S_2$

Distributed control synthesis

Splitting of the system

$$\dot{x}_1 = f_1(x_1, x_2, u_1)$$
$$\dot{x}_2 = f_2(x_1, x_2, u_2)$$

Control objective: (R, S)-stability with $R = R_1 \times R_2$, $S = S_1 \times S_2$

- Basic idea: (R_1, S_1)-stability synthesis for sub-system 1 by considering sub-system 2 as a bounded perturbation (in S_2) and vice-versa

Distributed control synthesis

Splitting of the system

$$\dot{x}_1 = f_1(x_1, x_2, u_1)$$
$$\dot{x}_2 = f_2(x_1, x_2, u_2)$$

Control objective: (R, S)-stability with $R = R_1 \times R_2$, $S = S_1 \times S_2$

- Basic idea: (R_1, S_1)-stability synthesis for sub-system 1 by considering sub-system 2 as a bounded perturbation (in S_2) and vice-versa
- Requirements:
  - Separated control (often possible)
  - Handling of bounded perturbations
  - Few interactions between sub-systems
  - Both syntheses successful

Principle of the distributed synthesis

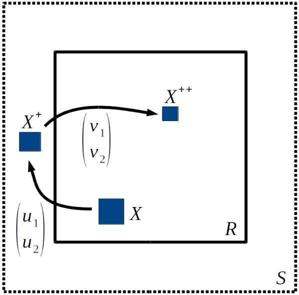
A discrete-time distributed control system:

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

Principle of the distributed synthesis

A discrete-time distributed control system:

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

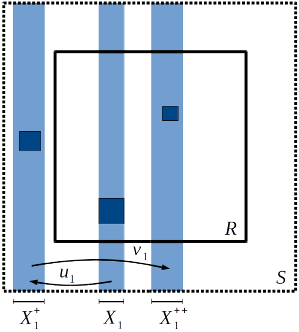


- $X \subset R$
- $X^{+} = f(X, \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}) \subset S$
- $X^{++} = f(X^{+}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}) \subset R$
- Pattern $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ depends on X

Principle of the distributed synthesis

A discrete-time distributed control system:

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



- $X_1 \subset R_1$
- $X_1^+ = f_1(X_1, S_2, u_1) \subset S_1$
- $X_1^{++} = f_1(X_1^+, S_2, v_1) \subset R_1$

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

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:

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

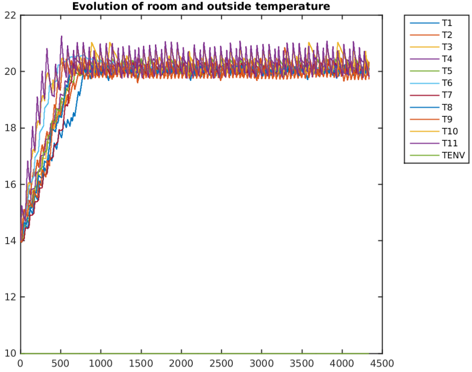
- System of dimension 11
- 2^{11} combinations of v_j (not all admissible, constraint on the number of open valves)
- Pipes heating a room may influence other rooms
- Doors opening and closing (here: average between open and closed)
- Varying external temperature (here: $T_{env} = 10^\circ C$)
- Measures 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:
$R = [18, 22]^{11}$
$S = [17.5, 22.5]^{11}$
$T_{env} = 10$

Output:
Iterated reachability in 15 steps
cpu time: 6h



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

Handling perturbations with the Euler scheme

Additional hypothesis on the dynamics:
(H_W): (Robustly OSL) $\exists\ \lambda_j \in \mathbb{R}$ and $\gamma_j \in \mathbb{R}_{\geq 0}$ s.t.

Handling perturbations with the Euler scheme

Additional hypothesis on the dynamics:
(H_W): (Robustly OSL) $\exists\ \lambda_j \in \mathbb{R}$ and $\gamma_j \in \mathbb{R}_{\geq 0}$ s.t.

$$\forall x, x' \in T,\ \forall w, w' \in W$$

$$\langle f_j(x,w) - f_j(x',w'), x - x' \rangle \leq \lambda_j \|x - x'\|^2 + \gamma_j \|x - x'\| \|w - w'\|.$$

NB: λ_j and γ_j can be computed with constrained optimization algorithms.
NB2: This notion is close to incremental input-to-state stability [Angeli].

Outline

1 Guaranteed control of switched systems

2 The reachable set computation

3 Distributed synthesis of controllers

4 **Control of partial differential equations with model order reduction**
- Model Order Reduction for high dimensional ODES
- Control of Partial Differential Equations

A high dimensional linear switched system with output

- Described by the differential equation:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases}$$

A high dimensional linear switched system with output

- Described by the differential equation:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases}$$

  - $x \in \mathbb{R}^n$: state variable
  - $y \in \mathbb{R}^m$: output
  - $u \in \mathbb{R}^p$: control input, takes a finite number of values (modes)
  - A,B,C: matrices of appropriate dimensions

A high dimensional linear switched system with output

- Described by the differential equation:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases}$$

  - $x \in \mathbb{R}^n$: state variable
  - $y \in \mathbb{R}^m$: output
  - $u \in \mathbb{R}^p$: control input, takes a finite number of values (modes)
  - A,B,C: matrices of appropriate dimensions

- Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$ by projection π_R (balanced truncation):

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) & = \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y_r(t) & = \hat{C}\hat{x}(t). \end{cases}$$

A high dimensional linear switched system with output

- Described by the differential equation:

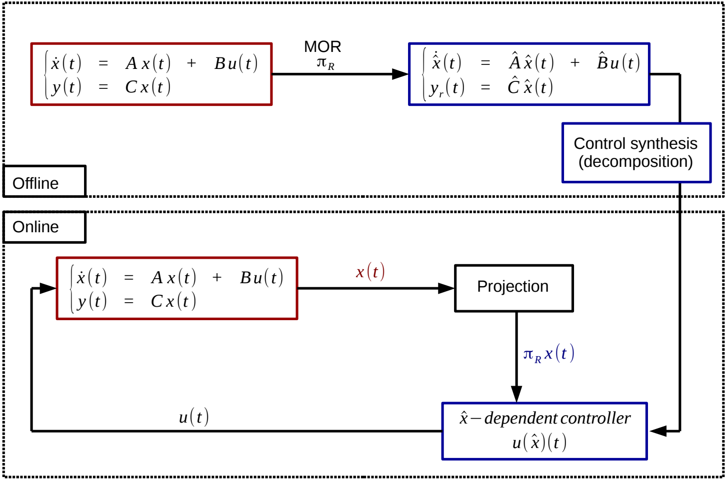
$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases}$$

  - $x \in \mathbb{R}^n$: state variable
  - $y \in \mathbb{R}^m$: output
  - $u \in \mathbb{R}^p$: control input, takes a finite number of values (modes)
  - A, B, C: matrices of appropriate dimensions
- Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$ by projection π_R (balanced truncation):

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) & = \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y_r(t) & = \hat{C}\hat{x}(t). \end{cases}$$

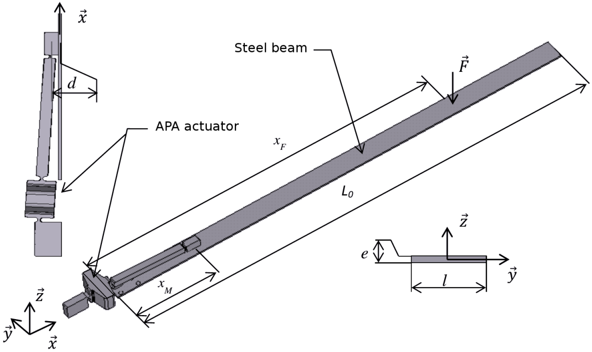
- Output trajectory error bound $\|y(t) - y_r(t)\|$
- Reduced state trajectory error bound $\|\pi_R x(t) - \hat{x}(t)\|$

Dealing with high dimensionality : Model Order Reduction



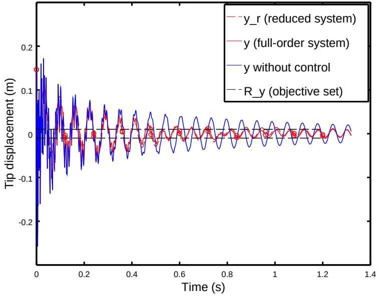
Application

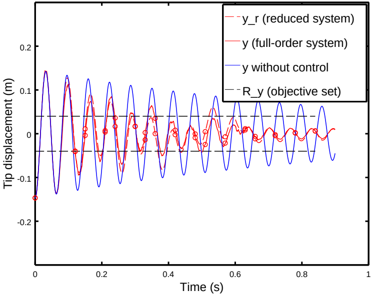
- Vibration (online) control of a cantilever beam:
  $n = 120$ and $n_r = 4$



Application

- Vibration (online) control of a cantilever beam:
  $n = 120$ and $n_r = 4$





Case of undiscretized PDE problems

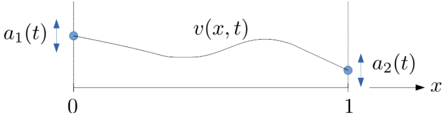
Difficulty:

- The problem becomes infinite-dimensional;
- Even spatially discretized, the *curse of dimensionality* makes the former approaches (bisection, ball overlapping, ...) irrelevant.

$\Longrightarrow$ requires model order reduction (MOR)

Pb of study: (ODE + 1D heat eq) with boundary control

$$\frac{d\boldsymbol{\xi}}{dt} = A_\sigma \boldsymbol{\xi} + \boldsymbol{b}_\sigma, \quad t > 0,$$
$$\frac{\partial u}{\partial t} - \nabla \cdot (\kappa(.)\nabla u) = f \quad \text{in } \Omega \times (0, +\infty),$$
$$u(0,t) = \xi_1(t), \quad u(L,t) = \xi_2(t), \quad \text{for all } t > 0,$$
$$u(.,t=0) = u^0$$



Use of 4 constant control modes:

$$\boldsymbol{b}_1 = (1,1)^T,\ \boldsymbol{b}_2 = (-1,-1)^T,\ \boldsymbol{b}_3 = (-1,1)^T,\ \boldsymbol{b}_4 = (1,-1)^T.$$

Control objective:

$$\boldsymbol{\xi}(t) \in R \quad \text{and} \quad \|u(.,t) - u^\infty\|_{L^2(0,1)} \leq \rho \quad \text{for all } t > 0.$$

Transformation of the problem

One can express the solution u as the sum of target state u^∞, quasistatic part of the solution u_q and a function ψ, i.e.

$$u(.,t) = u^\infty(.) + u_q(.,t) + \psi(.,t)$$

where $\psi(.,t)$ is solution of the heat problem with homogeneous Dirichlet boundary conditions. Look for a low dimensional approximation $\tilde{\psi}$ of ψ:

$$\tilde{\psi}(x,t) = \sum_{k=1}^{K} \tilde{\beta}_k(t)\varphi^k(x)$$

with a reduced basis $\{\varphi^k\}_{k=1,\dots,K}$ assumed to be orthonormal in $L^2(\Omega)$. Then

$$\|\psi(.,t)\|_{L^2(\Omega)} \leq \|\psi(.,t) - \tilde{\psi}(.,t)\|_{L^2(\Omega)} + \|\tilde{\beta}(t)\|_2.$$

Reduced order problem

Additional assumption (can be ensured by a proper construction of the reduced basis):

$$\|\psi(.,t)-\tilde{\psi}(.,t)\|_{L^2(\Omega)} \leq \mu\,\|\psi^0-\tilde{\psi}^0\|_{L^2(\Omega)} \quad \forall t \in [0,\tau]$$

Then continuity w.r.t. source term + maximum principle leads to:

Global stability requirement

$$C\,\|f+\nabla\cdot(\kappa(.)\nabla u^\infty)\|_{L^2(\Omega)} + L\,\|\xi(t)-\xi^\infty\|_\infty + \|\tilde{\beta}(t)\|_2 + \mu\,\|\psi^0-\pi^K\psi^0\|_{L^2(\Omega)} + \mu\,\|\beta^0-\tilde{\beta}^0\|_2 \leq \rho.$$

Numerical experiments

$$\frac{d\boldsymbol{a}}{dt} = \boldsymbol{b}_\sigma, \quad \boldsymbol{b}_\sigma \in \mathbb{R}^2,\ t > 0,$$

$$\alpha\, \partial_t v - \partial^2_{xx} v = 0 \quad \text{in } (0,1) \times (0,+\infty),$$

$$v(0,t) = a_1(t), \quad v(1,t) = a_2(t), \quad t > 0,$$

$$v(.,0) = v_0$$

- $K = 4$ (reduced-order space of dimension 2+4=6)
- max switching sequence length $= 8$
- Offline step: Overlapping of the stability domain by $4^6 = 4096$ balls, computed in less than 20 mins on a laptop
- Guaranteed control verified

Numerical experiments (2)

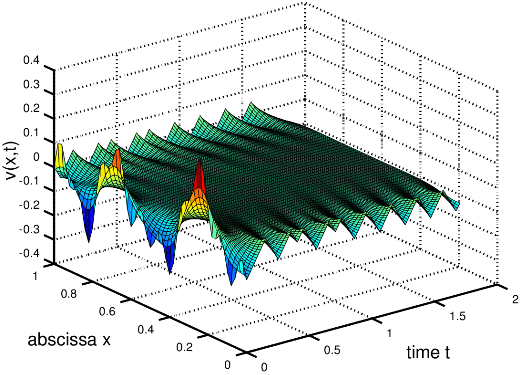


Figure: Controlled discrete solution $t \mapsto v(., t)$.

Conclusions

- Reachability analysis:
  - Guaranteed control of nonlinear switched systems using guaranteed RK4/Euler (without strong hypotheses)
  - Renewal of the Euler scheme using OSL property
  - Euler method: very fast but may lack accuracy
- Compositional synthesis allowing to handle higher dimensions:
  - Splitting system in sub-systems and local synthesis
  - Requires handling of (bounded) perturbations
- Control of PDEs made possible with Model Order Reduction
  - Discretized PDEs with projection methods
  - Undiscretized PDEs and proper transformation of the problem + Euler

Contributions

- New ways of performing reachability analysis without strong hypotheses
  $\Rightarrow$ 2 conference papers + 1 journal paper
- Distributed versions of synthesis algorithms
  $\Rightarrow$ 3 conference papers + 1 submitted journal paper
- Two procedures for control of PDEs
  $\Rightarrow$ 1 conference paper + 1 journal paper
- Algorithmic improvements and faster computation times
  $\Rightarrow$ 3 new branches for MINIMATOR (soon online) and C++ version

Perspectives

- Stochastic systems using Euler
- Guaranteed OSL constant
- Would the OSL property be relevant on other numerical schemes?
- Testing on concrete PDE case studies (SCOLE)
- Coupling of domain decomposition methods and compositional synthesis

Some references

Distributed control synthesis:

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

Distributed Control Synthesis Using Euler's Method, A. Le Coënt, J. Alexandre, A. Chapoutot, F. De Vuyst, L. Chamoin, L. Fribourg, *RP'17*.

Control of nonlinear systems using RK4:

Control of Nonlinear Switched Systems Based on Validated Simulation, A. Le Coënt, J. Alexandre dit Sandretto, A. Chapoutot and L. Fribourg, *SNR'16*.

Control of nonlinear systems using Euler and balls:

Control Synthesis of Nonlinear Sampled Switched Systems using Euler's Method, A. Le Coënt, F. De Vuyst, L. Chamoin, L. Fribourg, *SNR'17*.

Control of Partial Differential Equations:

Control of mechanical systems using set based methods, A. Le Coënt, F. De Vusyt, L. Chamoin, C. Rey and L. Fribourg, *International Journal of Dynamics and Control*.

Euler: 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:

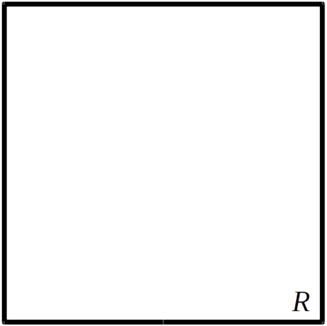
$$C_j = \sup_{x \in S} L_j \| f_j(x) \|$$

- Constant λ_j:

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

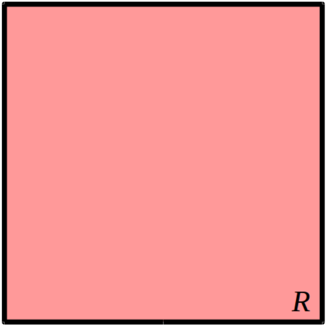
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



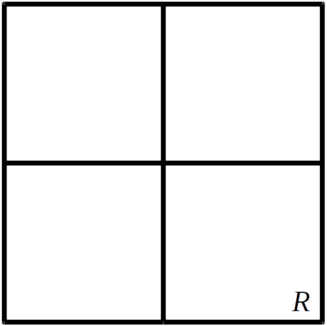
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



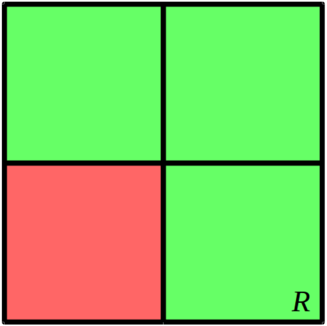
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



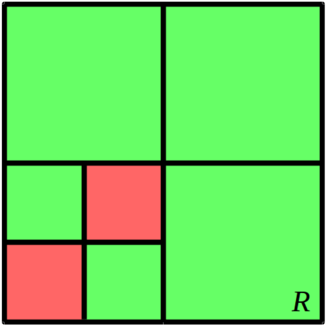
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



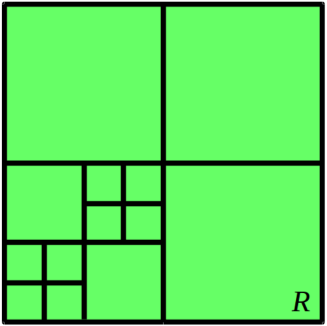
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



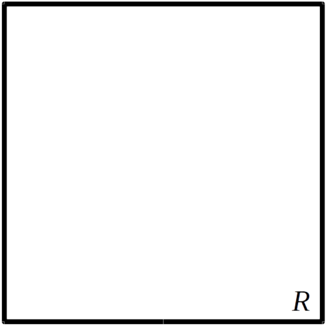
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of all the dimensions



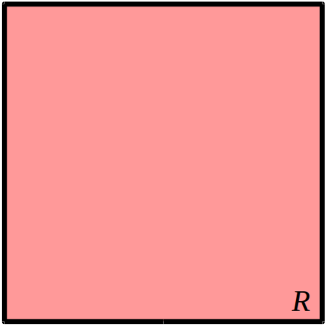
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



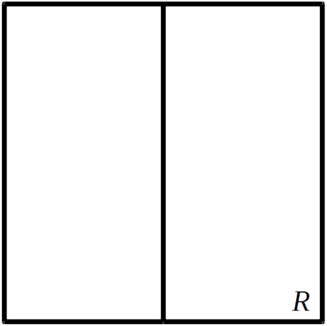
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



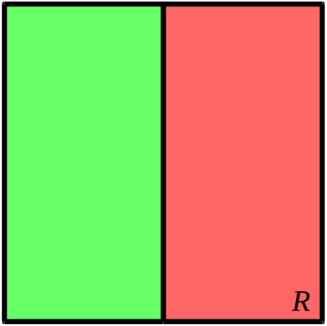
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



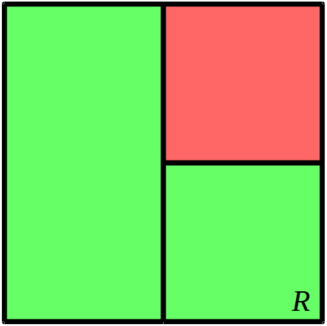
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



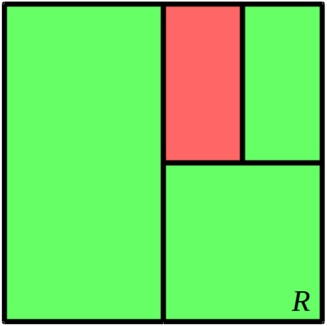
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



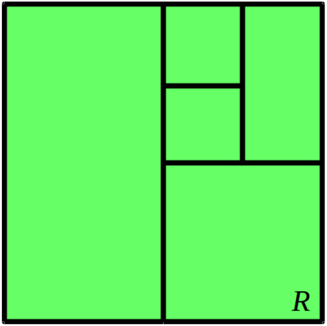
Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



Sate-space bisection algorithm: several heuristics possible

Heuristics: bisection of largest dimension



Euler: 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:

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

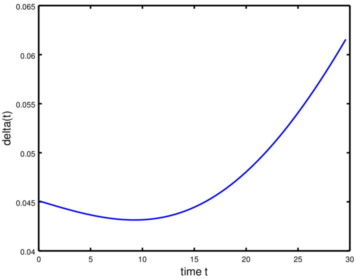
$$A_2 = \begin{pmatrix} -\frac{1}{x_l}(r_l + \frac{r_0 . r_c}{r_0+r_c}) & -\frac{1}{x_l}\frac{r_0}{r_0+r_c} \\ \frac{1}{x_c}\frac{r_0}{r_0+r_c} & -\frac{1}{x_c}\frac{r_0}{r_0+r_c} \end{pmatrix} \quad B_2 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

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

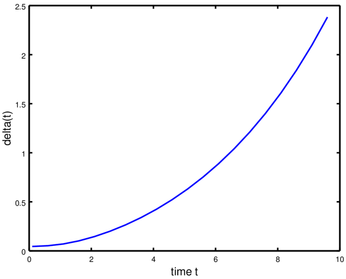
λ_1	-0.014215
λ_2	0.142474
C_1	6.7126×10^{-5}
C_2	2.6229×10^{-2}

Euler: Convexity of the trajectories

Example of a DC-DC converter:



$\lambda_1 = -0.014215$



$\lambda_2 = 0.142474$

HD-ODEs: Output and state trajectory error

After application of a pattern of length j

- the error between y and y_r is bounded by [Han-Krogh 2004]:

$$\varepsilon_y^j = \|u(\cdot)\|_\infty^{[0,j\tau]} \int_0^{j\tau} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt \quad + \\ \sup_{x_0 \in R_x} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau \hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|.$$

HD-ODEs: Output and state trajectory error

After application of a pattern of length j

- the error between y and y_r is bounded by [Han-Krogh 2004]:

$$\varepsilon_y^j = \|u(\cdot)\|_\infty^{[0,j\tau]} \int_0^{j\tau} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt \quad + \sup_{x_0 \in R_x} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau\hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|.$$

- the error between $\pi_R x$ and $\hat{x}$ is bounded by:

$$\varepsilon_x^j = \|u(\cdot)\|_\infty^{[0,j\tau]} \int_0^{j\tau} \| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt \quad + \sup_{x_0 \in R_x} \| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau\hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|.$$

PDEs: Transformation of the problem

Denoting by $u_q = u_q(., t)$ the solution of the quasi-static problem at each time t:

$$-\nabla \cdot (\kappa(.)\nabla u_q) = f + \nabla \cdot (\kappa(.)\nabla u^\infty) \text{ in } \Omega,$$
$$u_q(0, t) = \xi_1(t) - \xi_1^\infty,$$
$$u_q(L, t) = \xi_2(t) - \xi_2^\infty,$$

one can express the solution u as the sum of u^∞, u_q and a function ψ, i.e.

$$u(., t) = u^\infty(.) + u_q(., t) + \psi(., t)$$

where $\psi(., t)$ is solution of the heat problem with homogeneous Dirichlet boundary conditions

PDEs: Transformation of the problem

$\psi(.,t)$ is solution of the heat problem with homogeneous Dirichlet boundary conditions

$$\frac{\partial \psi}{\partial t} - \nabla \cdot (\kappa(.) \nabla \psi) = g(.;\boldsymbol{\xi}(t)) \quad \text{in } \Omega \times (0,+\infty)$$

$$\psi(0,t) = \psi(L,t) = 0, \quad t > 0,$$

$$\psi(.,t=0) = \psi^0,$$

with

$$g(.;\xi(t)) = -\frac{\partial u_q}{\partial t}(.;\boldsymbol{\xi}(t)), \quad \psi^0 = u^0 - u^\infty - u_q(.,0).$$

i.e., in variational form, find $\psi \in L^2(0,\infty;V)$, $\psi(.,t=0) = \psi^0$, such that

$$(\frac{\partial \psi}{\partial t}, v) + (\kappa(.)\nabla \psi, \nabla v) = (g(.;\boldsymbol{\xi}(t), v) \quad \forall v \in V.$$

PDEs: Transformation of the problem

Control objective:

$$\|u(.,t) - u^{\infty}(.)\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t > 0$$

With $u(.,t) = u^{\infty}(.) + u_q(.,t) + \psi(.,t)$, it becomes

$$\|u_q(.,t) + \psi(.,t)\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t > 0.$$

PDEs: Transformation of the problem

The solution u_q itself can be decomposed as

$$u_q(.,t) = \bar{u}(.) + w_q(.,t),$$

where $\bar{u}$ is solution of the steady elliptic problem with homogeneous Dirichlet boundary conditions

$$\begin{aligned} &-\nabla \cdot (\kappa(.)\bar{u}) = f + \nabla \cdot (\kappa(.)\nabla u^{\infty}) \quad \text{in } \Omega, \\ &\bar{u}(0) = \bar{u}(L) = 0, \end{aligned}$$

and w_q is solution of the quasistatic problem at each time t:

$$\begin{aligned} &-\nabla \cdot (\kappa(.)\nabla w_q) = 0 \text{ in } \Omega, \\ &w_q(0,t) = \xi_1(t) - \xi_1^{\infty}, \quad \text{for all } t > 0, \\ &w_q(L,t) = \xi_2(t) - \xi_2^{\infty}, \quad \text{for all } t > 0. \end{aligned}$$

PDEs: Transformation of the problem

The solution $\bar{u}$ is continuous with respect to the source term, i.e.

$$\|\bar{u}\|_V \leq C' \, \|f + \nabla \cdot (\kappa(.)\nabla u^\infty)\|_{L^2(\Omega)}.$$

For the solution w_q, because of the maximum principle, we have

$$\|w_q(.,t)\|_{L^\infty(\Omega)} = \max(|\xi_1(t) - \xi_1^\infty|, |\xi_2(t) - \xi_2^\infty|) = \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty.$$

Thus,

$$\begin{aligned} \|u_q(.,t) + \psi(.,t)\|_{L^2(\Omega)} \quad &\leq \quad \|\bar{u}\|_{L^2(\Omega)} + \|w_q\|_{L^2(\Omega)} + \|\psi(.,t)\|_{L^2(\Omega)} \\ &\leq \; C\|f + \nabla \cdot (\kappa(.)\nabla u^\infty)\|_{L^2(\Omega)} + L\,\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \|\psi(.,t)\|_{L^2(\Omega)} \end{aligned}$$

PDEs: Reduced order problem

Look for a low dimensional approximation $\tilde{\psi}$ of ψ:

$$\tilde{\psi}(x,t) = \sum_{k=1}^{K} \tilde{\beta}_k(t)\varphi^k(x)$$

with a reduced basis $\{\varphi^k\}_{k=1,\dots,K}$ assumed to be orthonormal in $L^2(\Omega)$. Then

$$\|\tilde{\psi}(.,t)\|_{L^2(\Omega)} = \|\tilde{\beta}(t)\|_{2,\mathbb{R}^K}.$$

By the triangular inequality we can write

$$\begin{aligned} \|\psi(.,t)\|_{L^2(\Omega)} &\leq \|\psi(.,t) - \tilde{\psi}(.,t)\|_{L^2(\Omega)} + \|\tilde{\psi}(.,t)\|_{L^2(\Omega)} \\ &\leq \|\psi(.,t) - \tilde{\psi}(.,t)\|_{L^2(\Omega)} + \|\tilde{\beta}(t)\|_2. \end{aligned}$$

PDEs: Reduced order problem

Additional assumption (can be ensured by a proper construction of the reduced basis):

$$\|\psi(.,t) - \tilde{\psi}(.,t)\|_{L^2(\Omega)} \leq \mu \, \|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \quad \forall t \in [0,\tau]$$

Then:

$$C\|f + \nabla \cdot (\kappa(.)\nabla u^\infty)\|_{L^2(\Omega)} + L\,\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \\ \|\tilde{\boldsymbol{\beta}}(t)\|_2 + \mu\,\|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \leq \rho.$$

And finally:

Global stability requirement

$$C\,\|f + \nabla \cdot (\kappa(.)\nabla u^\infty)\|_{L^2(\Omega)} + L\,\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \\ \|\tilde{\boldsymbol{\beta}}(t)\|_2 + \mu\,\|\psi^0 - \pi^K\psi^0\|_{L^2(\Omega)} + \mu\,\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^0\|_2 \leq \rho.$$

PDEs: Strategy for stability control

We look for control modes (or patterns) that ensure on $[0, k\tau]$ (before the next mode/pattern)

$$L\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^{\infty}\|_{\infty} \leq \delta_{\xi},$$

$$\|\tilde{\boldsymbol{\beta}}(t)\|_2 \leq \rho_{\beta},$$

$$\|\psi(t) - \tilde{\psi}(t)\|_{L^2(\Omega)} \leq \delta.$$

with

$$c_1 + \delta_{\xi} + \rho_{\beta} + \delta \leq \rho$$

and $c_1 = C\,\|f + \nabla \cdot (\kappa(.)\nabla u^{\infty})\|_{L^2(\Omega)}$.
Then by construction we will automatically fulfill the stability requirement on the heat solution for a given control mode σ, i.e.

$$\|u(., t) - u^{\infty}\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t \in (0, \tau].$$

PDEs possible application: the SCOLE model

$$\rho v_{tt}(x,t) + EIv_{xxxx}(x,t) + \rho B v_t(x,t) = \rho \omega^2 v(x,t)$$
$$v(0,t) = v_x(0,t) = v_{xx}(L,t) = v_{xxx}(L,t) = 0$$
$$\omega_t(t) = \frac{\Gamma(t) - 2\omega(t)\int_0^L \rho v(x,t) v_t(x,t) dx}{I_d + \int_0^L \rho v^2(x,t) dx}$$