High Dimensional Switched Systems: Control and Observation

Adrien Le Coënt [1], Florian De Vuyst [1], Christian Rey [2], Ludovic Chamoin [2], Laurent Fribourg [3], Mario Sigalotti[4]

October 14, 2015

[1]CMLA Centre de Mathématiques et de Leurs Applications
[2]LMT-Cachan Laboratoire de Mécanique et Technologie
[3]LSV Laboratoire de Spécification et Vérification
[4]CMAP Centre de Mathématiques Appliquées

Introduction

Framework

- Goal: control the evolution of an operating system with the help of actuators and sensors
- Framework of the switched control systems: one selects the working modes of the system over time, every mode is described by differential equations (ODEs or PDEs)
- Application to medium/high dimensional systems:
  - Model Order Reduction
  - Error bounding
  - State space bisection

Outline

Outline

Switched Systems

A switched system

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

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

Switched Systems

A switched system

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

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 switched system

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

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

- state $x \in \mathbb{R}^n$
- switching signal $\sigma : [0, \infty) \to U$

Switched Systems

A switched system

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

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

- state $x \in \mathbb{R}^n$
- switching signal $\sigma : [0, \infty) \to U$
- $U = \{1, \dots, N\}$ finite set of modes associated with the dynamics

$$\dot{x} = f_u(x)$$

Switched Systems

A switched system

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

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

- state $x \in \mathbb{R}^n$
- switching signal $\sigma : [0, \infty) \to U$
- $U = \{1, \dots, N\}$ finite set of modes associated with the dynamics

$$\dot{x} = Ax + Bu$$

Switched Systems

A switched system

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

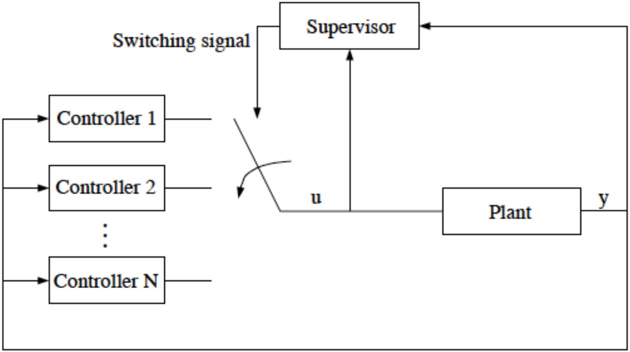
is a family of continuous-time dynamical systems with a rule σ that determines at each time which one is active

- state $x \in \mathbb{R}^n$
- switching signal $\sigma : [0, \infty) \to U$
- $U = \{1, \ldots, N\}$ finite set of modes associated with the dynamics

$$\dot{x} = Ax + Bu$$

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

Controlled Switched Systems: Schematic View



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 τ, 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 τ, find the appropriate switched mode $u \in U$ according to the current value of x, in order to achieve some objectives:

- stability (x should converge to and stay in the neighborhood R of a reference point Ω)

Control Synthesis Problem

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

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

- stability (x should converge to and stay in the neighborhood R of a reference point Ω)
- safety (x should never exit from a safe zone S)

Control Synthesis Problem

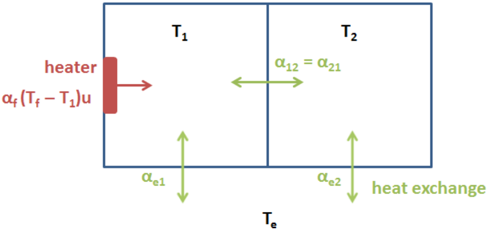
We consider the state-dependent control problem of synthesizing σ:

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

- stability (x should converge to and stay in the neighborhood R of a reference point Ω)
- safety (x should never exit from a safe zone S)

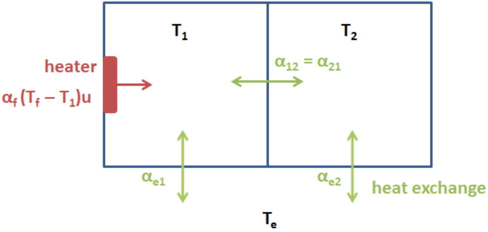
NB: classic stabilization impossible here (no common equilibrium pt)
$\rightsquigarrow$ practical stability

Example: Two-room apartment



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

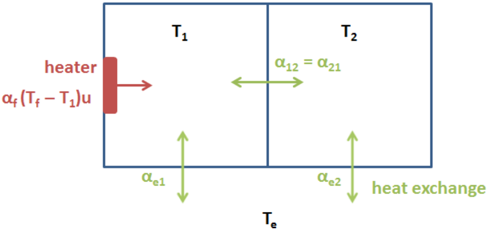
Example: Two-room apartment



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

- Modes: $u = 0, 1$; sampling period τ

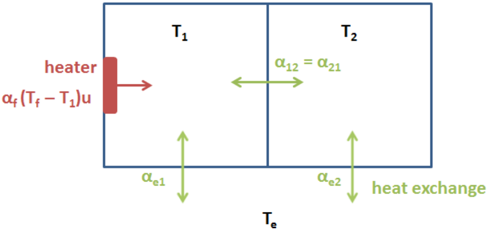
Example: Two-room apartment



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

- Modes: $u = 0, 1$; sampling period τ
- A pattern π is a finite sequence of modes (e.g. $(1 \cdot 0 \cdot 0 \cdot 0)$)

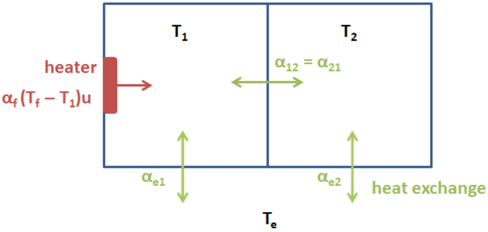
Example: Two-room apartment



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

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

Example: Two-room apartment



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

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

NB: Each mode has its basic proper equilibrium point; by appropriate switching, one can drive the system to a specific stability zone

Safety and Stability Properties for the two-room apartment

Safety and Stability Properties for the two-room apartment

- Example of safety property to be checked: satisfactory temperature

$$\forall t \geq 0 : \qquad T_{min} \leq T_i(t) \leq T_{max}$$

Safety and Stability Properties for the two-room apartment

- Example of safety property to be checked: satisfactory temperature

$$\forall t \geq 0 : \qquad T_{min} \leq T_i(t) \leq T_{max}$$

- Example of stability property to be checked: temperature regulation

$$|T_i(t) - T_{reference}| \leq \varepsilon \text{ as } t \to \infty$$

Outline

1 Switched Systems

2 State Space Decomposition

3 Control of high dimensional switched systems

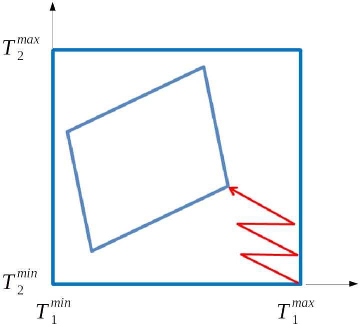
4 Observation of high dimensional switched systems

Bisection Method

Bisection Method

Given a zone R (selected around a reference point Ω of the state-space)

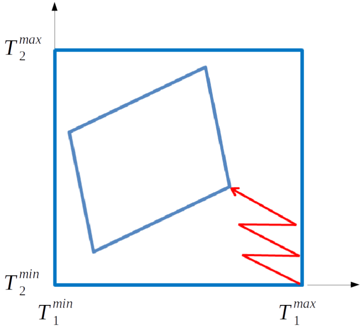
- Look for a *pattern* which maps R into R



Bisection Method

Given a zone R (selected around a reference point Ω of the state-space)

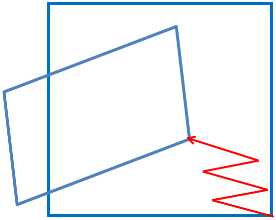
- Look for a *pattern* which maps R into R
- If such a *pattern* exists, then uniform control over the whole R



Bisection Method

Given a zone R (selected around a reference point Ω of the state-space)

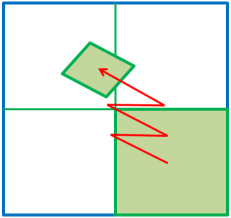
- Look for a *pattern* which maps R into R
- If such a *pattern* exists, then uniform control over the whole R
- Otherwise,



Bisection Method

Given a zone R (selected around a reference point Ω of the state-space)

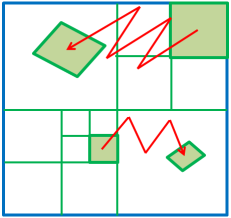
- Look for a *pattern* which maps R into R
- If such a *pattern* exists, then uniform control over the whole R
- Otherwise, bisect of R into subparts, and search for *patterns* mapping these subparts into R



Bisection Method

Given a zone R (selected around a reference point Ω of the state-space)

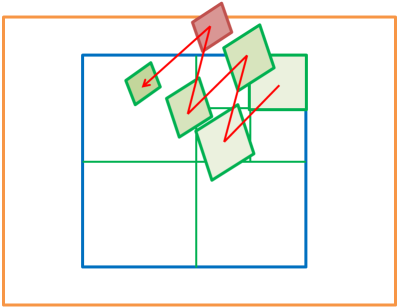
- Look for a *pattern* which maps R into R
- If such a *pattern* exists, then uniform control over the whole R
- Otherwise, bisect of R into subparts, and search for *patterns* mapping these subparts into R
- In case of failure, iterate the bisection

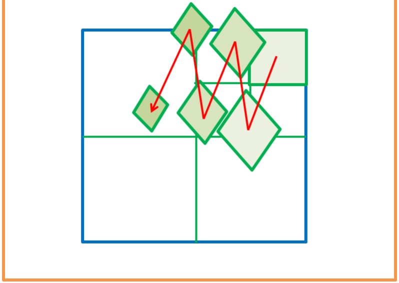


Bisection Method

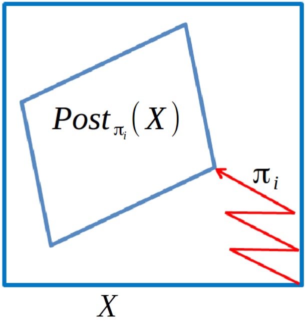
Given a zone R (selected around a reference point Ω of the state-space)

- Look for a *pattern* which maps R into R
- If such a *pattern* exists, then uniform control over the whole R
- Otherwise, bisect of R into subparts, and search for *patterns* mapping these subparts into R
- In case of failure, iterate the bisection
- Extension for safety: the unfolding must stay in the safety set S.





Post Set Operators



definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R$

definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R$
- $\forall i \in I \; Post_{\pi_i}(V_i) \subseteq R$

definition

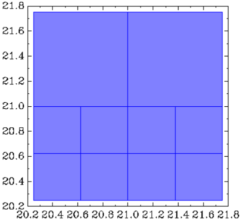
A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R$
- $\forall i \in I \ Post_{\pi_i}(V_i) \subseteq R$
- (Extension for safety: and $\forall i \in I \ Unf_{\pi_i}(V_i) \subseteq S$).

definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

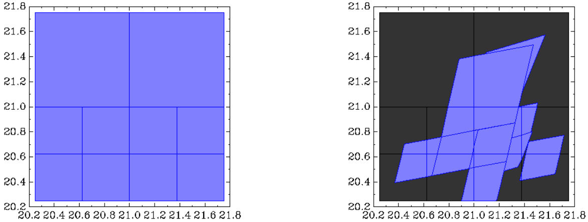
- $\bigcup_{i \in I} V_i = R$
- $\forall i \in I \; Post_{\pi_i}(V_i) \subseteq R$
- (Extension for safety: and $\forall i \in I \; Unf_{\pi_i}(V_i) \subseteq S$).



definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R$
- $\forall i \in I \; Post_{\pi_i}(V_i) \subseteq R$
- (Extension for safety: and $\forall i \in I \; Unf_{\pi_i}(V_i) \subseteq S$).



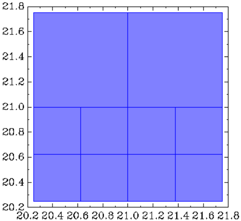
definition and property

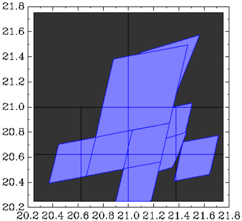
Let $Post_\Delta(X) =_{def} \bigcup_{i \in I} Post_{\pi_i}(X \cap V_i)$.

definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R$
- $\forall i \in I \ Post_{\pi_i}(V_i) \subseteq R$
- (Extension for safety: and $\forall i \in I \ Unf_{\pi_i}(V_i) \subseteq S$).





definition and property

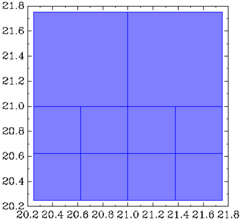
Let $Post_\Delta(X) =_{def} \bigcup_{i \in I} Post_{\pi_i}(X \cap V_i)$. We have:

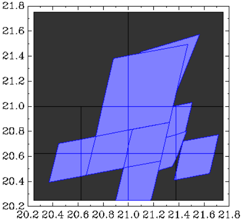
$$Post_\Delta(R) \subseteq R$$

definition

A decomposition Δ of R is a set of couples $\{(V_i, \pi_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R$
- $\forall i \in I \ Post_{\pi_i}(V_i) \subseteq R$
- (Extension for safety: and $\forall i \in I \ Unf_{\pi_i}(V_i) \subseteq S$).





definition and property

Let $Post_\Delta(X) =_{def} \bigcup_{i\in I} Post_{\pi_i}(X \cap V_i)$. We have:

$$Post_\Delta(R) \subseteq R \quad (\text{and} \quad Unf_\Delta(R) \subseteq S)$$

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

1. $x(t) \in R$, therefore $\exists i \in I$ such that $x(t) \in V_i$

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

1. $x(t) \in R$, therefore $\exists i \in I$ such that $x(t) \in V_i$
2. Apply pattern π_i to $x(t)$

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

1. $x(t) \in R$, therefore $\exists i \in I$ such that $x(t) \in V_i$
2. Apply pattern π_i to $x(t)$
3. At the end of π_i, $x(t') \in R$, iterate by going back to step (1)

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

1. $x(t) \in R$, therefore $\exists i \in I$ such that $x(t) \in V_i$
2. Apply pattern π_i to $x(t)$
3. At the end of π_i, $x(t') \in R$, iterate by going back to step (1)

Property

Under the Δ-control,

- any trajectory $x_0 \rightarrow_{\pi_{i_1}} x_1 \rightarrow_{\pi_{i_2}} x_2 \rightarrow_{\pi_{i_3}} \cdots$ always stays in R

Control and Trajectories Induced by Δ

The decomposition $\Delta = \{(V_i, \pi_i)\}_{i \in I}$ induces a natural control:

1. $x(t) \in R$, therefore $\exists i \in I$ such that $x(t) \in V_i$
2. Apply pattern π_i to $x(t)$
3. At the end of π_i, $x(t') \in R$, iterate by going back to step (1)

Property

Under the Δ-control,

- any trajectory $x_0 \rightarrow_{\pi_{i_1}} x_1 \rightarrow_{\pi_{i_2}} x_2 \rightarrow_{\pi_{i_3}} \cdots$ always stays in R
- The unfolding of the trajectory always stays in S

Decomposition for the two-room apartment

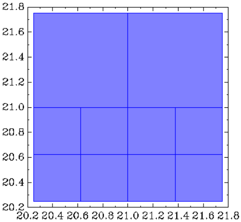
For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.

Decomposition for the two-room apartment

For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$

Decomposition for the two-room apartment

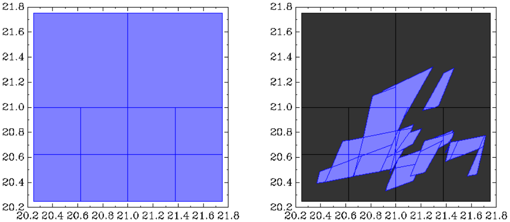
For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$



Decomposition for the two-room apartment

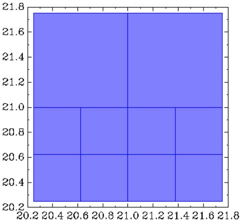
For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.

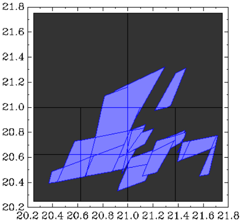
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$

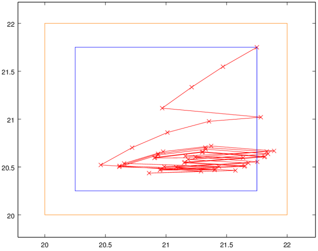


Decomposition for the two-room apartment

For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$







Decomposition for the two-room apartment

For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$

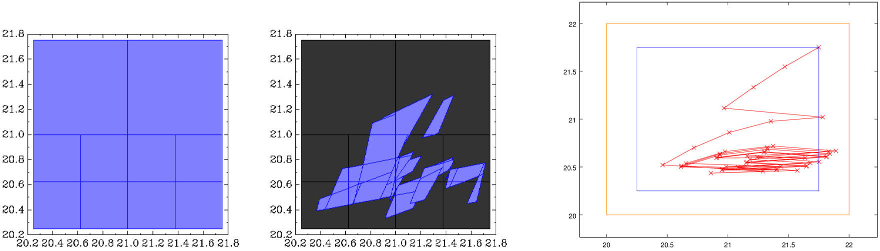


Figure : Decomposition (left) ; unfolding (middle) ; unfolded trajectory (right) in plane (T_1, T_2)

Decomposition for the two-room apartment

For: $\alpha_{12} = 5 \times 10^{-2}, \alpha_{21} = 5 \times 10^{-2}, \alpha_{e1} = 5 \times 10^{-3}, \alpha_{e2} = 3.3 \times 10^{-3}, \alpha_f = 8.3 \times 10^{-3}, T_e = 10, T_f = 50$ and $\tau = 5$.
$\Omega = (21, 21)$, $R = [20.25, 21.75] \times [20.25, 21.75]$, $S = [20, 22] \times [20, 22]$

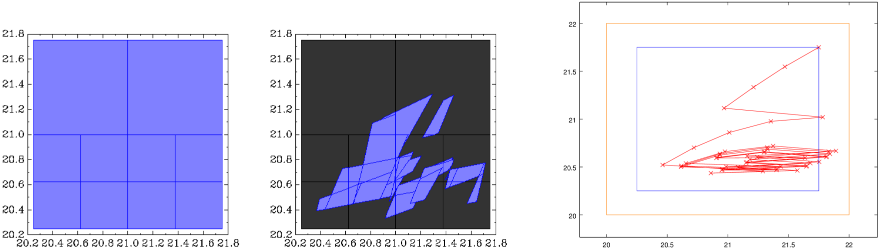


Figure : Decomposition (left) ; unfolding (middle) ; unfolded trajectory (right) in plane (T_1, T_2)

Decomposition found for $k = 4, d = 3$.

A 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 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 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
- Idea: impose the right $u(t)$ such that x and y verify some properties (stability, reachability...)

A 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
- Idea: impose the right $u(t)$ such that x and y verify some properties (stability, reachability...)
- Objectives:
  1. *x-stabilization*: make all the state trajectories starting in a compact interest set $R_x \subset \mathbb{R}^n$ return to R_x;
  2. *y-convergence*: send the output of all the trajectories starting in R_x into an objective set $R_y \subset \mathbb{R}^m$;

A Switched System with Output

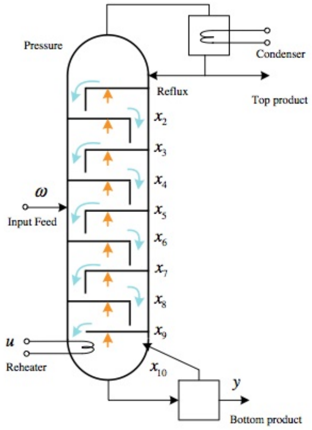
- Described by the differential equation:

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

  - $x \in \mathbb{R}^n$: 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
- Idea: impose the right $u(t)$ such that x and y verify some properties (stability, reachability...)
- Objectives:
  1. *x-stabilization*: make all the state trajectories starting in a compact interest set $R_x \subset \mathbb{R}^n$ return to R_x;
  2. *y-convergence*: send the output of all the trajectories starting in R_x into an objective set $R_y \subset \mathbb{R}^m$;
- Constraint: x of “high” dimension.

A Sampled Switched System with Output

A distillation column



New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i) \subseteq R_x$ (x-stabilization)

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i) \subseteq R_x$ (x-stabilization)
- $\forall i \in I \; Post_{Pat_i, C}(V_i) \subseteq R_y$ (y-convergence)

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i) \subseteq R_x$ (x-stabilization)
- $\forall i \in I \; Post_{Pat_i, C}(V_i) \subseteq R_y$ (y-convergence)

definition and property

Let $Post_\Delta(X) =_{def} \bigcup_{i \in I} Post_{\pi_i}(X \cap V_i)$.

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i) \subseteq R_x$ (x-stabilization)
- $\forall i \in I \; Post_{Pat_i, C}(V_i) \subseteq R_y$ (y-convergence)

definition and property

Let $Post_\Delta(X) =_{def} \bigcup_{i \in I} Post_{\pi_i}(X \cap V_i)$. We have:

$$Post_\Delta(R_x) \subseteq R_x \quad \text{and} \quad Post_{\Delta, C}(R_x) \subseteq R_y.$$

New Decomposition

definition

A decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i) \subseteq R_x$ (x-stabilization)
- $\forall i \in I \; Post_{Pat_i,C}(V_i) \subseteq R_y$ (y-convergence)

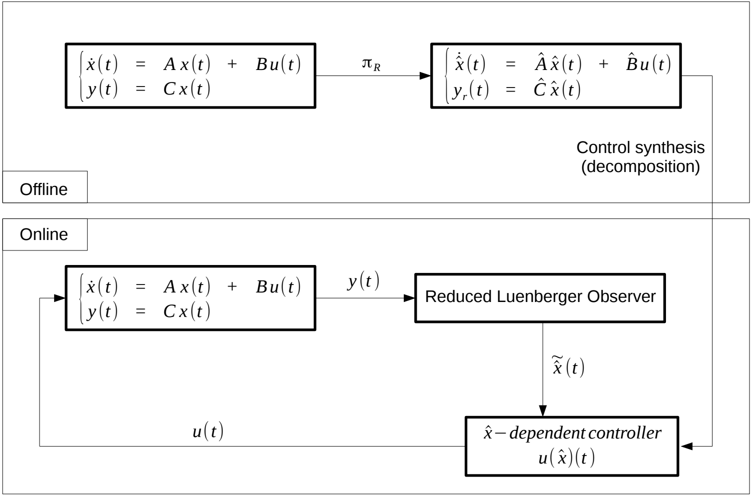
definition and property

Let $Post_\Delta(X) =_{def} \bigcup_{i\in I} Post_{\pi_i}(X \cap V_i)$. We have:

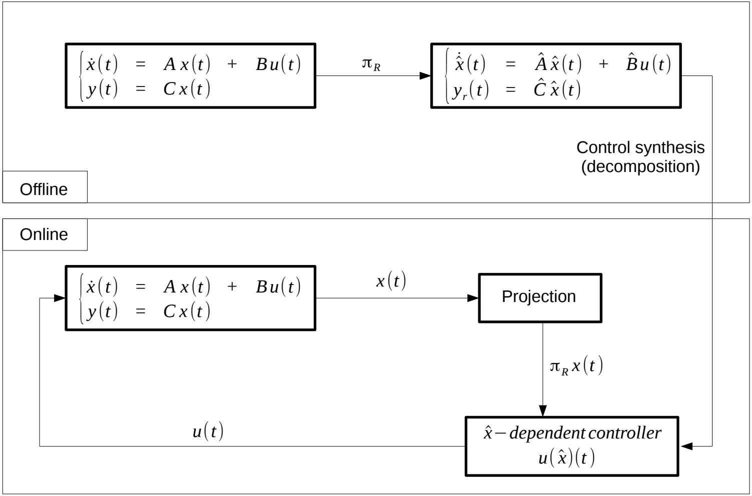
$$Post_\Delta(R_x) \subseteq R_x \quad \text{and} \quad Post_{\Delta,C}(R_x) \subseteq R_y.$$

Computational cost of decomposition: at most in $O(2^{nd}N^k)$.

Dealing with high dimensionality : model reduction



Dealing with high dimensionality : model reduction



Outline

1. Switched Systems
2. State Space Decomposition
3. Control of high dimensional switched systems
   - Model Order Reduction
   - Guaranteed offline control
   - Guaranteed online control
4. Observation of high dimensional switched systems

Outline

Model Order Reduction by Projection

Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$:

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

Model Order Reduction by Projection

Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$:

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

Reduction by a projection (constructed by balanced truncation) $\pi = \pi_L \pi_R$, $\pi_L \in \mathbb{R}^{n \times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$:

$$\hat{A} = \pi_R A \pi_L, \quad \hat{B} = \pi_R B, \quad \hat{C} = C \pi_L.$$

Model Order Reduction by Projection

Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$:

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

Reduction by a projection (constructed by balanced truncation) $\pi = \pi_L \pi_R$, $\pi_L \in \mathbb{R}^{n \times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$:

$$\hat{A} = \pi_R A \pi_L, \quad \hat{B} = \pi_R B, \quad \hat{C} = C\pi_L.$$

Goal: design a controle rule $u(\cdot)$ at the low-order level and apply it at the full-order level.

Model Order Reduction by Projection

Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$:

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

Reduction by a projection (constructed by balanced truncation) $\pi = \pi_L \pi_R$, $\pi_L \in \mathbb{R}^{n \times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$:

$$\hat{A} = \pi_R A \pi_L, \quad \hat{B} = \pi_R B, \quad \hat{C} = C\pi_L.$$

Goal: design a controle rule $u(\cdot)$ at the low-order level and apply it at the full-order level.

Requirements:

- projection of the interest set $\hat{R}_x = \pi_R R_x$

Model Order Reduction by Projection

Construction of a reduced order system $\hat{\Sigma}$ of order $n_r < n$:

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

Reduction by a projection (constructed by balanced truncation) $\pi = \pi_L \pi_R$, $\pi_L \in \mathbb{R}^{n \times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$:

$$\hat{A} = \pi_R A \pi_L, \quad \hat{B} = \pi_R B, \quad \hat{C} = C\pi_L.$$

Goal: design a controle rule $u(\cdot)$ at the low-order level and apply it at the full-order level.
Requirements:

- projection of the interest set $\hat{R}_x = \pi_R R_x$
- error bounding of the state and output trajectory

Output and state trajectory error [2]

After application of a pattern of length j

- the error between y and y_r is bounded by:

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

Output and state trajectory error [2]

After application of a pattern of length j

- the error between y and y_r is bounded by:

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

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

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

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Control synthesis (decomposition) for the reduced-order system.

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Control synthesis (decomposition) for the reduced-order system.
$\Rightarrow$ reduced-order control

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Control synthesis (decomposition) for the reduced-order system.

$\Rightarrow$ reduced-order control

$\Rightarrow$ application of the reduced-order control to the full-order system

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Control synthesis (decomposition) for the reduced-order system.

$\Rightarrow$ reduced-order control

$\Rightarrow$ application of the reduced-order control to the full-order system

Questions:

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

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

Control synthesis (decomposition) for the reduced-order system.

$\Rightarrow$ reduced-order control

$\Rightarrow$ application of the reduced-order control to the full-order system

Questions:

- How is it applied?

Reduced Order Control

Two systems:

- Full-order system: Σ, R_x, R_y

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

- Reduced-order system: $\hat{\Sigma}$, $\hat{R}_x$, R_y

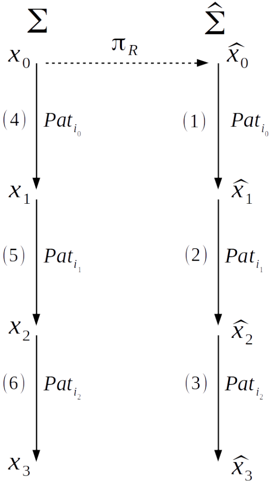
$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) & = \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y_r(t) & = \hat{C}\hat{x}(t). \end{cases}$$

Control synthesis (decomposition) for the reduced-order system.
$\Rightarrow$ reduced-order control
$\Rightarrow$ application of the reduced-order control to the full-order system
Questions:

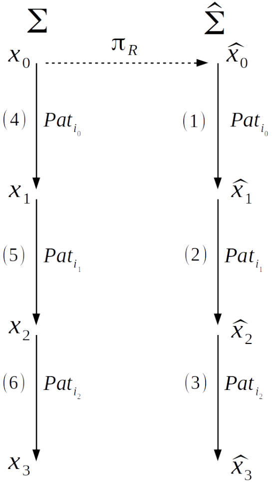
- How is it applied?
- Is the reduced-order control effective at the full-order level?

Outline

Offline Procedure

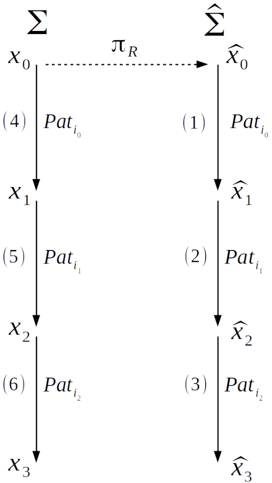


Offline Procedure



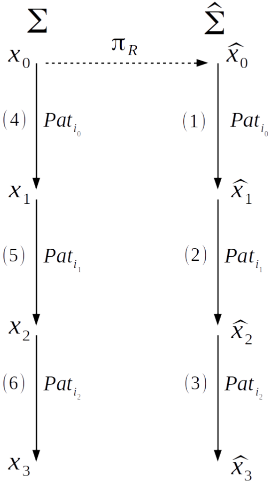
1. Projection of the initial state x_0

Offline Procedure



1. Projection of the initial state x_0
2. Computation of a pattern sequence at the low-order level Pat_{i_0}, Pat_{i_1}... (steps (1),(2),(3))

Offline Procedure



1. Projection of the initial state x_0
2. Computation of a pattern sequence at the low-order level Pat_{i_0}, Pat_{i_1}... (steps (1),(2),(3))
3. Application of the pattern sequence at the full-order level (steps (4),(5),(6)).

Guaranteed Offline Control

Application of the same pattern sequence:

Guaranteed Offline Control

Application of the same pattern sequence:

$$\Rightarrow \quad \forall t = j\tau > 0, \quad \|y(t) - y_r(t)\| \leq \varepsilon_y^j$$

Guaranteed Offline Control

Application of the same pattern sequence:

$$\Rightarrow \quad \forall t = j\tau > 0, \quad \|y(t) - y_r(t)\| \leq \varepsilon_y^j$$

$$\Rightarrow \quad \forall t = j\tau > 0, \quad \|y(t) - y_r(t)\| \leq \varepsilon_y^\infty = \sup_{j>0} \varepsilon_y^j$$

Guaranteed Offline Control

Application of the same pattern sequence:

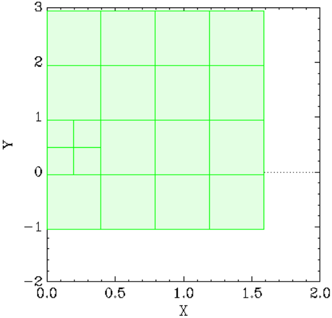
$$\Rightarrow \quad \forall t = j\tau > 0, \quad \|y(t) - y_r(t)\| \leq \varepsilon_y^j$$

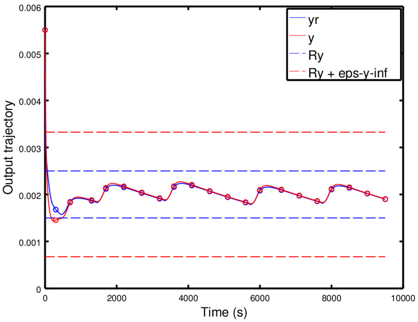
$$\Rightarrow \quad \forall t = j\tau > 0, \quad \|y(t) - y_r(t)\| \leq \varepsilon_y^\infty = \sup_{j>0} \varepsilon_y^j$$

Consequence: the output of the full order system is sent in $R_y + \varepsilon_y^\infty$.

Guaranteed Offline Control

Simulation on a linearized model of a distillation column: $n = 11$ and $n_r = 2$:

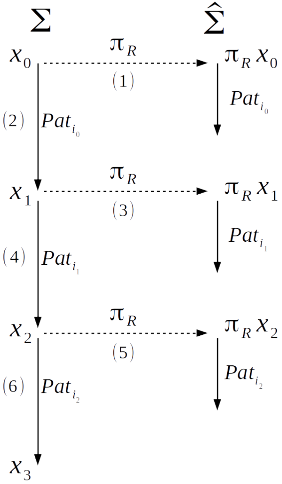




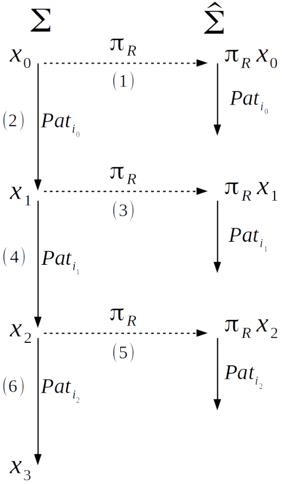
Outline

1 Switched Systems
2 State Space Decomposition
3 Control of high dimensional switched systems
   - Model Order Reduction
   - Guaranteed offline control
   - Guaranteed online control
4 Observation of high dimensional switched systems
   - Observation of switched systems
   - Numerical test of a reduced order observer

Online Procedure

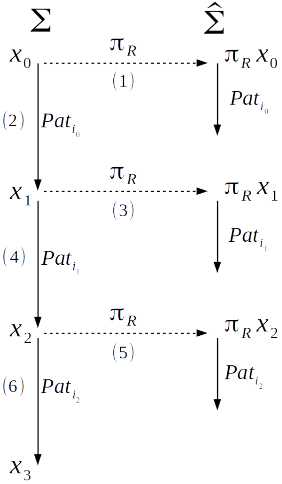


Online Procedure



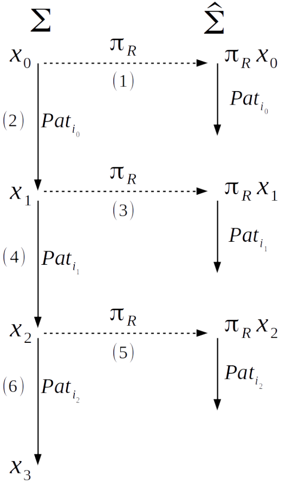
1. Projection of the initial state x_0 (step (1))

Online Procedure



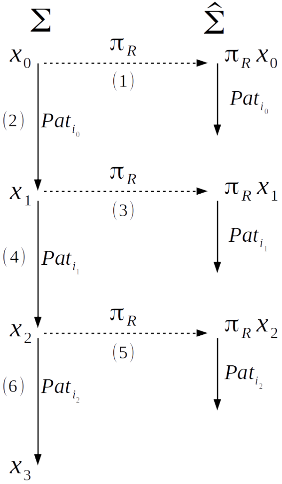
1. Projection of the initial state x_0 (step (1))
2. Computation of the pattern Pat_{i_0} at the reduced-order level

Online Procedure



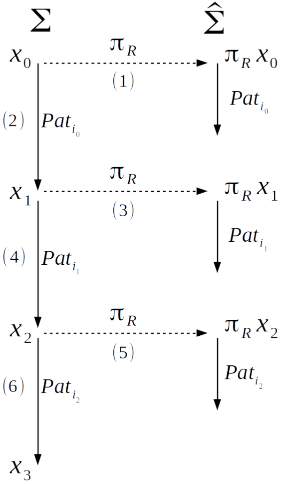
1. Projection of the initial state x_0 (step (1))
2. Computation of the pattern Pat_{i_0} at the reduced-order level
3. Application of the pattern Pat_{i_0} at the full-order level, Σ is sent to a state x_1 (step (2))

Online Procedure



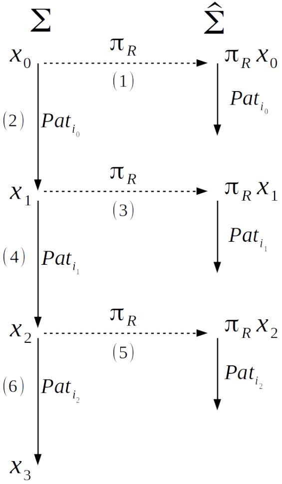
1. Projection of the initial state x_0 (step (1))
2. Computation of the pattern Pat_{i_0} at the reduced-order level
3. Application of the pattern Pat_{i_0} at the full-order level, Σ is sent to a state x_1 (step (2))
4. Projection of the (new initial) state x_1 (step (3))

Online Procedure



1. Projection of the initial state x_0 (step (1))
2. Computation of the pattern Pat_{i_0} at the reduced-order level
3. Application of the pattern Pat_{i_0} at the full-order level, Σ is sent to a state x_1 (step (2))
4. Projection of the (new initial) state x_1 (step (3))
5. Computation of the pattern Pat_{i_1} at the reduced-order level

Online Procedure



1. Projection of the initial state x_0 (step (1))
2. Computation of the pattern Pat_{i_0} at the reduced-order level
3. Application of the pattern Pat_{i_0} at the full-order level, Σ is sent to a state x_1 (step (2))
4. Projection of the (new initial) state x_1 (step (3))
5. Computation of the pattern Pat_{i_1} at the reduced-order level
6. Application of the pattern Pat_{i_1} at the full-order level, Σ is sent to a state x_2 (step (4))...

Guaranteed Online Control

Requirement to apply the online procedure:

Guaranteed Online Control

Requirement to apply the online procedure:

- Ensure that $\pi_R Post_{Pat_i}(x) \in \hat{R}_x$ at every step.

Guaranteed Online Control

Requirement to apply the online procedure:

- Ensure that $\pi_R Post_{Pat_i}(x) \in \hat{R}_x$ at every step.

Solution: Compute an ε-decomposition

definition

A ε-decomposition Δ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$
- $\forall i \in I \ Post_{Pat_i}(V_i) \subseteq R_x - \varepsilon_x^{|Pat_i|}$
- $\forall i \in I \ Post_{Pat_i,C}(V_i) \subseteq R_y$ (y-convergence)

Guaranteed Online Control

An ε-decomposition performed on $\hat{\Sigma}$ permits to iterate the online procedure:

Guaranteed Online Control

An ε-decomposition performed on $\hat{\Sigma}$ permits to iterate the online procedure:

- At a step k, $\pi_R x_k$ is sent in $\hat{R}_x - \varepsilon_x^{|Pat_{i_k}|}$

Guaranteed Online Control

An ε-decomposition performed on $\hat{\Sigma}$ permits to iterate the online procedure:

- At a step k, $\pi_R x_k$ is sent in $\hat{R}_x - \varepsilon_x^{|Pat_{i_k}|}$
- we have:

$$\|\pi_R Post_{Pat}(x) - Post_{Pat}(\pi_R x)\| \leq \varepsilon_x^{|Pat_{i_k}|}$$

Guaranteed Online Control

An ε-decomposition performed on $\hat{\Sigma}$ permits to iterate the online procedure:

- At a step k, $\pi_R x_k$ is sent in $\hat{R}_x - \varepsilon_x^{|Pat_{i_k}|}$
- we have:

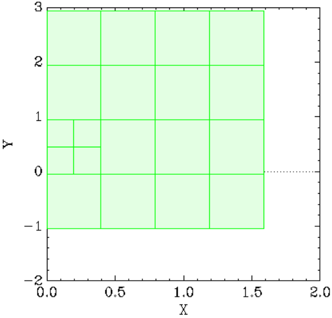
$$\|\pi_R Post_{Pat}(x) - Post_{Pat}(\pi_R x)\| \leq \varepsilon_x^{|Pat_{i_k}|}$$

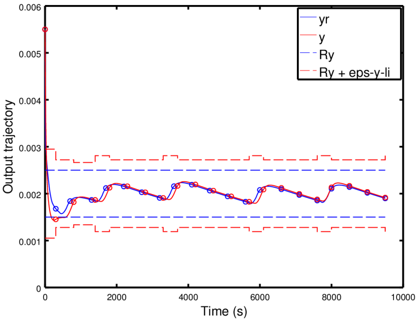
- thus, at every step k:

$$\pi_R Post_{Pat_{i_k}}(x_k) \in \hat{R}_x$$

Guaranteed Online Control

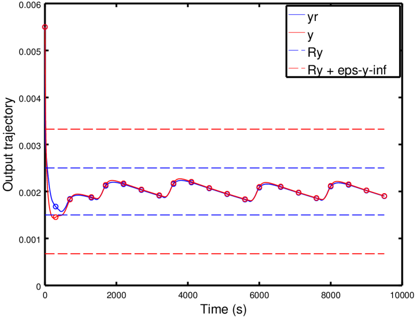
Simulation on a linearized model of a distillation column: $n = 11$ and $n_r = 2$:

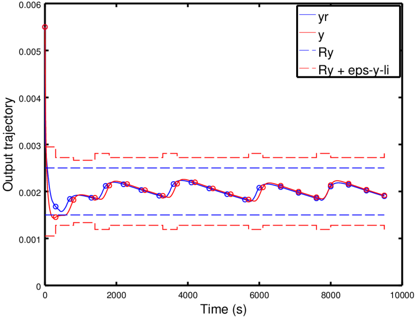




Remark: Output trajectory error depending on the length of the applied pattern: much lower than the infinite bound ε_y^∞

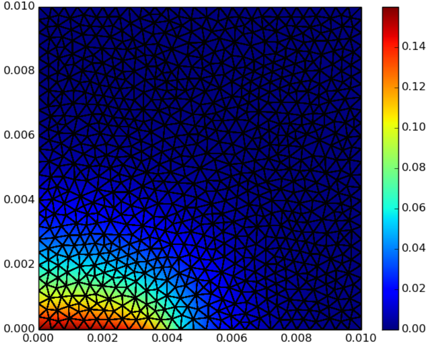
Comparison of the Two Procedures





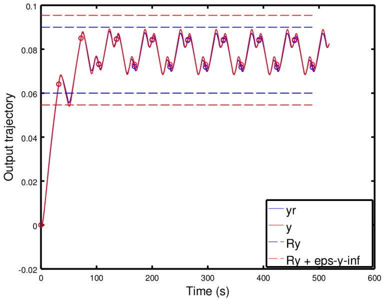
Other Applications

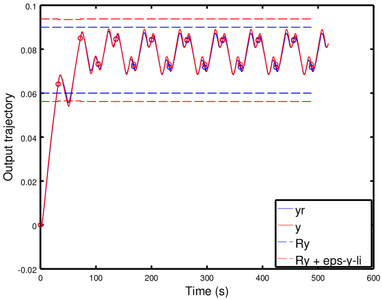
- Control of the temperature of a square plate discretized by finite elements: offline and online control
  $n = 897$



Other Applications

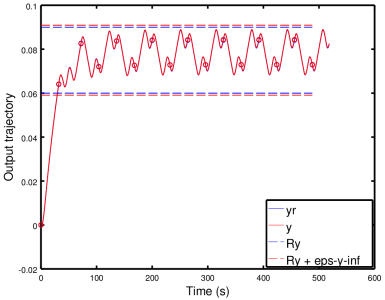
- Control of the temperature of a square plate discretized by finite elements: offline and online control
  $n = 897$ and $n_r = 2$

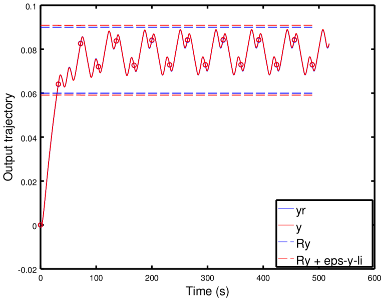




Other Applications

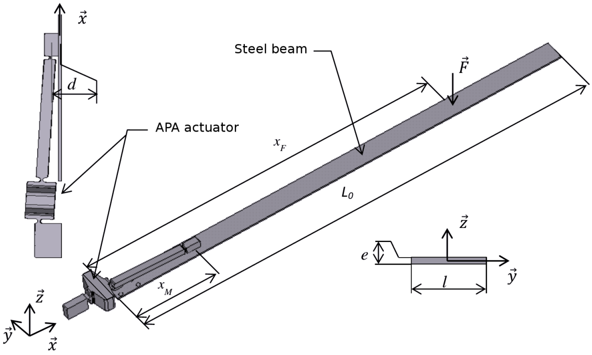
- Control of the temperature of a square plate discretized by finite elements: offline and online control
  $n = 897$ and $n_r = 3$





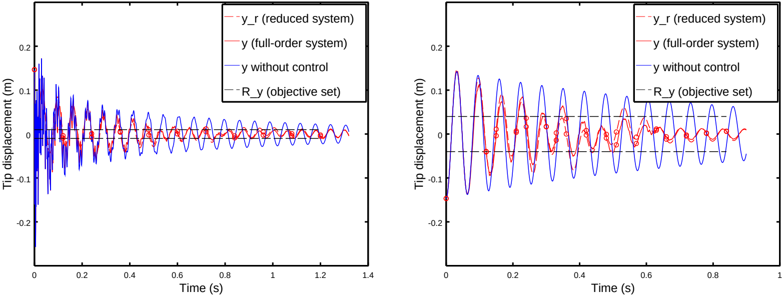
Other Applications

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



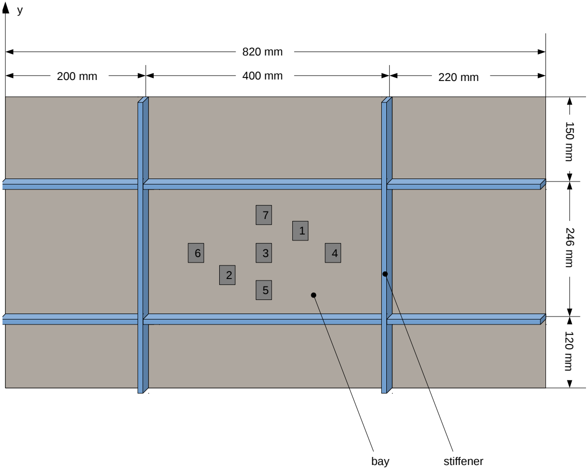
Other Applications

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



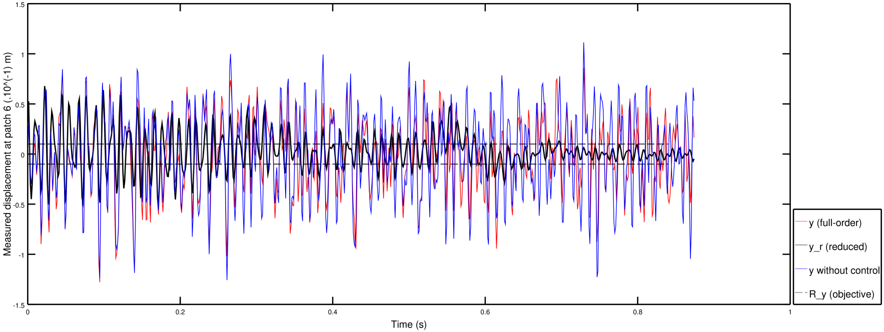
Other Applications

- Vibration (online) control of an aircraft panel:
  $n = 57000$ and $n_r = 6$



Other Applications

- Vibration (online) control of an aircraft panel: $n = 57000$ and $n_r = 6$



Outline

Outline

Partial observation (without model reduction)

Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.

Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.
Question: how can we control Σ with the only information of y ?

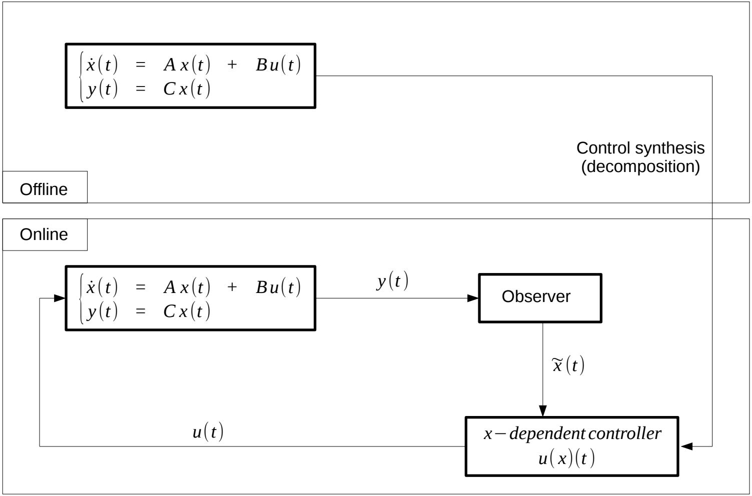
Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.
Question: how can we control Σ with the only information of y ?
$\Rightarrow$ An observer: intermediate system, provides an estimate $\tilde{x}$ of the state x of Σ

Partial observation (without model reduction)



Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.
Question: how can we control Σ with the only information of y ?
$\Rightarrow$ An observer: intermediate system, provides an estimate $\tilde{x}$ of the state x of Σ

Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.
Question: how can we control Σ with the only information of y ?
$\Rightarrow$ An observer: intermediate system, provides an estimate $\tilde{x}$ of the state x of Σ
Question: which observer ?

Partial observation (without model reduction)

Given the switched system:

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

During a real online use, only $y(t)$ is known.
Question: how can we control Σ with the only information of y ?
$\Rightarrow$ An observer: intermediate system, provides an estimate $\tilde{x}$ of the state x of Σ
Question: which observer ?
$\Rightarrow$ Kalman filter, High gain observer, Luenberger observer?

Why the Luenberger observer?

- Dynamics of the Luenberger observer:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n \times m}$$

Why the Luenberger observer?

- Dynamics of the Luenberger observer:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n \times m}$$

$\Rightarrow$ Adapted to switched systems (because of $L(u)$)

Why the Luenberger observer?

- Dynamics of the Luenberger observer:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n\times m}$$

$\Rightarrow$ Adapted to switched systems (because of $L(u)$)

$\Rightarrow$ Easy implementation

Why the Luenberger observer?

- Dynamics of the Luenberger observer:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n\times m}$$

$\Rightarrow$ Adapted to switched systems (because of $L(u)$)

$\Rightarrow$ Easy implementation

$\Rightarrow$ Many good properties...

Why the Luenberger observer?

- Dynamics of the Luenberger observer:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n\times m}$$

$\Rightarrow$ Adapted to switched systems (because of $L(u)$)

$\Rightarrow$ Easy implementation

$\Rightarrow$ Many good properties...

- Objective: find a strategy such that the observer converges: $\eta(t) = |\tilde{x}(t) - x(t)| \underset{t\to+\infty}{\longrightarrow} 0$

Properties of the Luenberger observer

Hypotheses:

- $\exists P > 0, \quad s.t. \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0 \quad \forall u.$
- (Dwell-time: $\tau > 0$)

Properties of the Luenberger observer

Hypotheses:

- $\exists P > 0, \quad s.t. \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0 \quad \forall u.$
- (Dwell-time: $\tau > 0$)

Theorem
[Serres, Vivalda, Riedinger, IEEE Trans.Auto.Cont. 2011]

With an appropriate[1] choice of patterns, the observer converges

[1] appropriate = every pattern takes particular values given by the study of e.

Properties of the Luenberger observer

Hypotheses:

- $\exists P > 0, \quad s.t. \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0 \quad \forall u.$
- (Dwell-time: $\tau > 0$)

Theorem
[Serres, Vivalda, Riedinger, IEEE Trans.Auto.Cont. 2011]

With an appropriate[1] choice of patterns, the observer converges monotonically.

[1] appropriate = every pattern takes particular values given by the study of e.

Properties of the Luenberger observer

Hypotheses:

- $\exists P > 0, \quad s.t. \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0 \quad \forall u.$
- (Dwell-time: $\tau > 0$)

Theorem

[Serres, Vivalda, Riedinger, IEEE Trans.Auto.Cont. 2011]

With an appropriate[1] choice of patterns, the observer converges monotonically.
i.e. $\eta(t) \underset{t \to +\infty}{\longrightarrow} 0$ and $\eta(t)$ decreases monotonically.

[1] appropriate = every pattern takes particular values given by the study of e.

Properties of the Luenberger observer

Hypotheses:

- $\exists P > 0, \quad s.t. \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0 \quad \forall u.$
- (Dwell-time: $\tau > 0$)

Theorem
[Serres, Vivalda, Riedinger, IEEE Trans.Auto.Cont. 2011]

With an appropriate[1] choice of patterns, the observer converges monotonically.
i.e. $\eta(t) \underset{t \to +\infty}{\longrightarrow} 0$ and $\eta(t)$ decreases monotonically.

Proof based on the study of

$$\dot{e} = (A - L(u)C)e$$

where $e(t) = x(t) - \tilde{x}(t)$

[1] appropriate = every pattern takes particular values given by the study of e.

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i \in I}$ such that:

- $\bigcup_{i \in I} V_i = R_x$

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \ Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$ and Pat_i takes particular value

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$ and Pat_i takes particular value
- $\forall i \in I \; Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$ and Pat_i takes particular value

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \ Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$ and Pat_i takes particular value
- $\forall i \in I \ Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$ and Pat_i takes particular value

definition and property

Let $Post_{\Delta}(X) =_{def} \bigcup_{i\in I} Post_{\pi_i}(X \cap V_i)$.

Observer based decomposition

Supposing that the initial reconstruction error is inferior to η_0

definition

A observer based decomposition $\tilde{\Delta}$ of R_x is a set of couples $\{(V_i, Pat_i)\}_{i\in I}$ such that:

- $\bigcup_{i\in I} V_i = R_x$
- $\forall i \in I \; Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$ and Pat_i takes particular value
- $\forall i \in I \; Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$ and Pat_i takes particular value

definition and property

Let $Post_\Delta(X) =_{def} \bigcup_{i\in I} Post_{\pi_i}(X \cap V_i)$. We have:

$$Post_{\tilde{\Delta}}(R_x + \eta_0) \subseteq R_x - \eta_0 \quad \text{and} \quad Post_{\Delta,C}(R_x + \eta_0) \subseteq R_y.$$

Outline

Numerical implementation with model reduction

An ε-decomposition is performed.

Use of a reduced Luenberger observer:

$$\dot{\tilde{\hat{x}}} = \hat{A}\tilde{\hat{x}} - L(u)(\hat{C}\tilde{\hat{x}} - Cx) + \hat{B}u, \quad L(u) \in \mathbb{R}^{n_r \times m}$$

Numerical implementation with model reduction

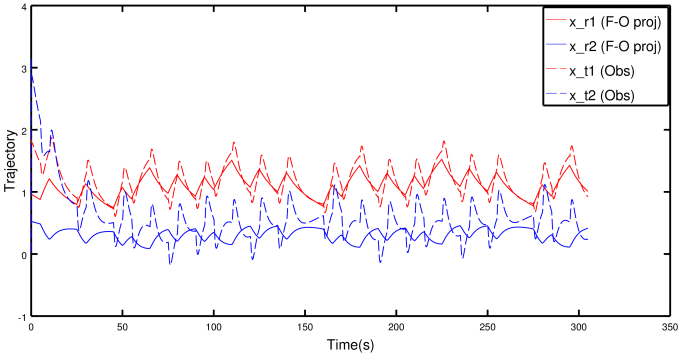
An ε-decomposition is performed.

Use of a reduced Luenberger observer:

$$\dot{\hat{\tilde{x}}} = \hat{A}\hat{\tilde{x}} - L(u)(\hat{C}\hat{\tilde{x}} - Cx) + \hat{B}u, \quad L(u) \in \mathbb{R}^{n_r \times m}$$

Simulation on the thermal plate problem:

Full-order system initialized at 0.06^{897}, observer initialized at 0^{897}



Numerical implementation with model reduction

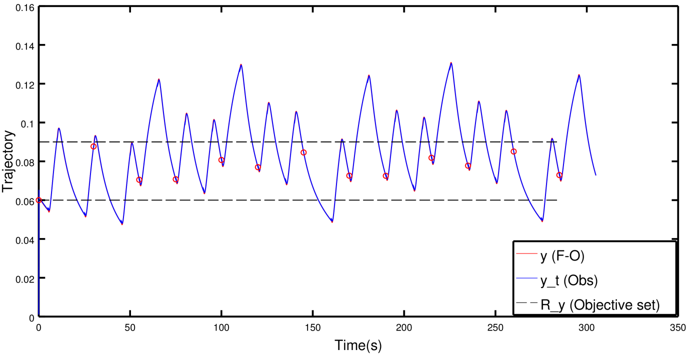
An ε-decomposition is performed.

Use of a reduced Luenberger observer:

$$\dot{\hat{\tilde{x}}} = \hat{A}\hat{\tilde{x}} - L(u)(\hat{C}\hat{\tilde{x}} - Cx) + \hat{B}u, \quad L(u) \in \mathbb{R}^{n_r \times m}$$

Simulation on the thermal plate problem:

Full-order system initialized at 0.06^{897}, observer initialized at 0^{897}



Conclusions

- Guaranteed reduced order control
- Guaranteed observer based control
- Numerical simulations encouraging for reduced observer based control, but no proof of the efficiency yet (ingredient required: a bound of the error between $\pi_R x$ and $\tilde{\hat{x}}$, W.I.P.)

Conclusions

- Guaranteed reduced order control
- Guaranteed observer based control
- Numerical simulations encouraging for reduced observer based control, but no proof of the efficiency yet (ingredient required: a bound of the error between $\pi_R x$ and $\tilde{\hat{x}}$, W.I.P.)

Future work

- Decomposition using dimensionality reduction (projection on more adapted reduced spaces using post-process techniques)
- Improvement of model reduction techniques (adapted to hyperbolic and non-linear systems)
- Control of non-linear systems/PDEs

Some References

Laurent Fribourg, Ulrich Kühne, and Romain Soulat.
Minimator: a tool for controller synthesis and computation of minimal invariant sets for linear switched systems, March 2013.

Zhi Han and Bruce Krogh.
Reachability analysis of hybrid systems using reduced-order models.
In *American Control Conference*, pages 1183–1189. IEEE, 2004.

Adrien Le Coënt, Florian de Vusyt, Christian Rey, Ludovic Chamoin, and Laurent Fribourg.
Guaranteed control of switched control systems using model order reduction and state-space bisection.
Open Acces Series in Informatics, 2015.

Ulysse Serres, Jean-Claude Vivalda, and Pierre Riedinger.
On the convergence of linear switched systems.
IEEE Transactions on Automatic Control, 56(2):320–332, 2011.

Some References

Laurent Fribourg, Ulrich Kühne, and Romain Soulat.
Minimator: a tool for controller synthesis and computation of minimal invariant sets for linear switched systems, March 2013.

Zhi Han and Bruce Krogh.
Reachability analysis of hybrid systems using reduced-order models.
In *American Control Conference*, pages 1183–1189. IEEE, 2004.

Adrien Le Coënt, Florian de Vusyt, Christian Rey, Ludovic Chamoin, and Laurent Fribourg.
Guaranteed control of switched control systems using model order reduction and state-space bisection.
Open Acces Series in Informatics, 2015.

Ulysse Serres, Jean-Claude Vivalda, and Pierre Riedinger.
On the convergence of linear switched systems.
IEEE Transactions on Automatic Control, 56(2):320–332, 2011.

Thank you ! Questions?