International Journal of Dynamics and Control manuscript No.
(will be inserted by the editor)

Control of Mechanical Systems Using Set Based Methods

Adrien Le Coënt · Florian de Vuyst · Christian Rey · Ludovic Chamoin · Laurent Fribourg

Received: date / Accepted: date

Abstract This paper considers large discrete-time linear systems obtained from discretized partial differential equations, and controlled by a *quantized* law, i.e., a piecewise constant time function taking a finite set of values. We show how to generate the control by, first, applying *model reduction* to the original system, then using a "state-space bisection" method for synthesizing a control at the reduced-order level, and finally computing an upper bound on the deviations between the controlled output trajectories of the reduced-order model and those of the original model. The effectiveness of our approach is illustrated on several examples of the literature.

A. Le Coënt
CMLA, ENS Cachan, CNRS, Université Paris-Saclay
61 av. du Président Wilson, 94235 Cachan cedex
Tel.: +33147405900
Fax: +33147405901
E-mail: adrien.le-coent@ens-cachan.fr

F. de Vuyst
CMLA, ENS Cachan, CNRS, Université Paris-Saclay
61 av. du Président Wilson, 94235 Cachan cedex
E-mail: devuyst@cmla.ens-cachan.fr

C. Rey
Safran Tech
1 Rue GeneviÃÍve AubÃÏ, 78772 Magny les Hameaux, France
E-mail: christian.rey@safran.fr

L. Chamoin
LMT Cachan, ENS Cachan, CNRS, Université Paris-Saclay
61 av. du Président Wilson, 94235 Cachan cedex
E-mail: chamoin@lmt.ens-cachan.fr

L. Fribourg
LSV, ENS Cachan, CNRS, Université Paris-Saclay
61 av. du Président Wilson, 94235 Cachan cedex
E-mail: fribourg@lsv.ens-cachan.fr

Keywords Model Order Reduction · guaranteed control · stability control · error bounding · Luenberger observer

1 Introduction

We focus here on switched control systems, a class of hybrid systems recently used with success in various domains such as automotive industry and power electronics. While these models are usually used for (low dimensional) ordinary differential equations (ODEs) controlled with a piecewise constant function, it is also possible to use these models for control of mechanical systems. Indeed, the dynamics of most mechanical systems can be modeled by partial differential equations, and the spacial discretization of such systems leads to high dimensional ODEs. Controlled with a piecewise constant function on the boundary, and written in a proper way (the state space representation), one obtains high dimensional switched control systems. Several strategies have been developed to design control laws for such systems; however the associated algorithms are very expensive and require a limited state space dimension, so that a model order reduction is required in order to synthesize a controller at the reduced-order level. Here, the invariance analysis [15, 14, 16] is used to synthesize controllers, an offline and an online procedure are proposed to apply the controller to the full-order system. Offline and online controls are developed, and the computation of upper bounds of the error induced by the reduction allows to guarantee the effectiveness of the controller. Note that this paper is an extension of the conference paper [26], it includes some missing parts, more test cases, and the applications belong to the field of mechanics.

Comparison with related work. Model order reduction techniques for hybrid or switched systems are classically used in *numerical simulation* in order to construct, at the reduced level, trajectories which cannot be computed directly at the original level due to complexity and large size dimension [4,10]. Model reduction is used in order to perform *set-based reachability analysis* in [21]. Isolated trajectories issued from isolated points are not constructed, but (an over-approximation of) the infinite set of trajectories issued from a dense set of initial points. This allows to perform formal verification of properties such as *safety*. In both approaches, the control is *given* as an input of the problem. In contrast here, the control is *synthesized* using set-based methods in order to achieve by construction properties such as *convergence* and *stability*.

The problem of control synthesis for hybrid and switched systems has been widely studied and various tools exist. The Multi-Parametric Toolbox (MPT 3.0 [23]) for example solves optimal control problems using operations on polytopes. Most approaches make use of Lyapunov or the so-called "multiple Lyapunov functions" to solve the problem of control synthesis for switched systems - see for example [38]. The approximate bisimulation approach abstracts switched systems under the form of a discrete model [17,18] under certain Lyapunov-based stability conditions. The latter approach has been implemented in PESSOA [27] and CoSyMA [30]. The approach used in this paper avoids using Lyapunov functions and relies on the notion of "(controlled) invariant" [9].

While the latter approaches are mostly used for the control of low order ODEs, the control of mechanical systems can be realized using the control theory approach, where a continuous control law is guessed and proved to be efficient on the continuous PDE model [37,5]. The damping of vibration with piezoelectric devices is in particular a widely developed branch of the control of mechanical systems. The shunting of piezoelectric devices with electric circuits permits to convert the vibration energy into electric energy, which is then dissipated in the electric circuits [20]. Note that this approach can be active or passive, depending on the electric energy furnished to the electric circuit. A switched control approach is developed in [11,32], the piezoelectric device is shunted on several electric circuits, but only one is selected at a time depending on the state of the mechanical system. This approach is called semi-active since the electric circuits are passive but the switching requires energy. In the present paper, the approach is fully active.

Plan.

In Section 2, we give some preliminaries on switched control systems and their link with PDEs and mechanical systems. In Section 3, we introduce some elements of control theory and the state-space bisection method. In Section 4, we explain how to construct a reduced model, apply the state-space bisection method at this level, and compute upper bounds to the error induced at the original level. In Section 5, we propose two methods of control synthesis allowing to synthesize (either offline or online) a controller at the reduced-order level and apply it to the full-order system. In Section 6, we apply our approach to several examples of the literature. In section 7, we extend our method to the use of observers. We conclude in Section 8.

2 Background

We consider systems governed by Partial Differential Equations (PDEs) having actuators allowing to impose forces on the boundary; these systems can represent transient thermal problems, vibration problems... By applying the right external force at the right time, one can drive the system to a desired operating mode. Our goal here is to synthesize a law which, given the state of the system, computes the boundary force to apply.

In order to illustrate our approach, we use the example of the heat equation:

$$\begin{cases} \dfrac{\partial T}{\partial t}(x,t) - \alpha \Delta T(x,t) = 0 \ \forall (t,x) \in [0,T] \times \Omega \\ T(x,\cdot) = T^d(x,\cdot) \qquad \forall x \in \partial\Omega^T \\ \dfrac{\partial T}{\partial x}(x,\cdot).n = \varphi^d(x,\cdot) \qquad \forall x \in \partial\Omega^\varphi \\ T(x,0) = T_0(x) \end{cases} \tag{1}$$

Discretized by finite elements, the nodal temperatures $\{T\}$ are computed with respect to time, and the system becomes:

$$\begin{cases} C_{FE}\{\dot{T}\} + K_{FE}\{T\} = \{F^d\} \\ \{T(0)\} = \{T_0\} \end{cases} \tag{2}$$

The purpose is then to compute the forces $\{F^d\}$ with respect to time such that the temperature field verifies some desired properties.

For example, one may want to impose that the temperature in a particular node stays within a given temperature range. Usually, the quantities of interest one wants to control are given in discrete points, which are for example sensor measurements, or they are given as local averaging. Here, we consider the case where the quantities of interest can be directly extracted from the nodal values with a matrix called *output matrix* (see equation (3)).

We consider a particular kind of actuators; the force applied only takes a finite number N of values. For example, in equation (1) for the case of a room heated with a heater, the flux φ^d is equal to 0 when the heater is turned off and equal to a positive value when it is turned on. The control systems associated to such behaviors are called *switched control systems*, and this is exactly the framework we place ourselves in. In control theory, such dynamical systems are written under the following form, called *state-space representation*:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \tag{3}$$

The n-vector x is called the state of the system, the p-vector u is the control input, the m-vector y is the output of the system, A is an $n \times n$-matrix, B an $n \times p$-matrix, and C an $m \times n$ matrix. Writing the discretized equation (2) under this form is straightforward by multiplying the first line by C_{FE}^{-1} (which is invertible), and the state vector is then $\{T\}$. In the case of higher order PDEs (for example in the case of the wave equation), we merely need to enlarge the state vector to take the first derivative of the nodal values in it.

3 Some Elements of Control Theory

An algorithm of control synthesis for switched control systems has been developed in [14, 16]. This algorithm, called *state-space decomposition algorithm*, allows the computation of control laws for low dimensional Ordinary Differential Equations.

In order to use this algorithm for the control of high dimensional discretized PDEs, we first give some preliminary notions and results of control theory.

3.1 Preliminaries

As we place ourselves in the framework of switched control systems, the control variable u of a given system Σ takes its values in a finite set U. The elements of U are called the *switching modes.* The algorithm developed allows to compute a law $u(x)$ that permits to verify some desired properties. This means that, knowing the state x of the system, one knows the switching mode to apply in order to verify the given properties. Such a law is thus called *state-dependent.* Note that the switching modes are applied during a time τ, and thus the law $u(x)$ gives the switching modes to apply at the times $k\tau$ with $k \in \mathbb{N}$. The type of control law we want to apply can be schematized in Figure 1.

The entries of the problem are the following:

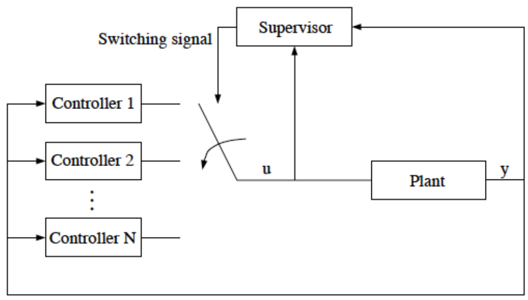

Fig. 1 Scheme of a switched control system

1. a subset $R_x \subset \mathbb{R}^n$ of the state space, called *interest set*,
2. a subset $R_y \subset \mathbb{R}^m$ of the output space, called *objective set.*

The objective is to find a law $u(\cdot)$ which, for any initial state $x_0 \in R_x$, stabilizes the output y in the set R_y. The set R_x is in fact the set of all the initial conditions considered, and the set R_y is a target set, where we want the output to stabilize. The sets R_x and R_y are given under the form of boxes, i.e. interval products of $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively.

We now introduce some notations and definitions required to present the algorithm. We will use $\mathbf{x}(t, x, u)$ to denote the point reached by Σ at time t under switching mode $u \in U$ from the initial condition x. This gives a transition relation $\rightarrow_u^\tau$ defined for x and x' in $\mathbb{R}^n$ by: $x \rightarrow_u^\tau x'$ iff $\mathbf{x}(\tau, x, u) = x'$. Given a set $X \subset \mathbb{R}^n$, we define the *successor set* of a set $X \subset \mathbb{R}^n$ of states under switching mode u as:

$$Post_u(X) = \{x' \mid x \rightarrow_u^\tau x' \text{ for some } x \in X\}.$$

As the systems considered are linear, the set $Post_u(X)$ is the result of the affine transformation $A_d X + B_d u$, where $A_d = e^{A\tau}, \quad B_d = \int_0^\tau e^{A(\tau - t)} B dt$. Likewise, we define the *output successor set* of a set $X \subset \mathbb{R}^n$ of states under switching mode u as:

$$Post_{u,C}(X) = \{Cx' \mid x \rightarrow_u^\tau x' \text{ for some } x \in X\}.$$

An *input pattern* named Pat is defined as a finite sequence of switching modes. A *k-pattern* is an input pattern of length at most k. The successor set of $X \subset \mathbb{R}^n$ using $Pat \equiv (u_1 \cdots u_k)$ is defined by

$$Post_{Pat}(X) = \{x' \mid x \rightarrow_{u_1}^\tau \cdots \rightarrow_{u_k}^\tau x', \ x \in X\}.$$

The mapping $Post_{Pat}$ is itself an affine transformation. The output successor set of $X \subset \mathbb{R}^n$ using $Pat \equiv (u_1 \cdots u_k)$ is defined by

$$Post_{Pat,C}(X) = \{Cx' \mid x \rightarrow_{u_1}^\tau \cdots \rightarrow_{u_k}^\tau x', \ x \in X\}.$$

3.2 State-Space Decomposition Algorithm

With these definitions and notations, we are now able to present the algorithm of control synthesis. It relies on the decomposition of the set R_x. Given the sets R_x and R_y, and a maximum length of input pattern K, it returns a set Δ of the form $\{(V_i, Pat_i)\}_{i \in I}$ where I is a finite set of indexes. Every V_i is a subset of R_x and every Pat_i is a k-pattern, such that:

(a) $\bigcup_{i \in I} V_i = R_x$,
(b) for all $i \in I$: $Post_{Pat_i}(V_i) \subseteq R_x$,
(c) for all $i \in I$: $Post_{Pat_i,C}(V_i) \subseteq R_y$.

The algorithm thus returns several sets V_i that cover R_x, and every V_i is associated to a pattern Pat_i that sends V_i in R_x, and the output in R_y. The set R_x is thus decomposed in several sets, and for each one, we have one control law: $\forall x \in V_i, u(x) = Pat_i$. Therefore, for two initial conditions in a set V_i, we apply the same input pattern. The fact that we use set based operations has a key role which allows us to consider sets of initial conditions, and this is how we manage to obtain a law $u(x)$. In the following, when a decomposition Δ is successfully obtained , we denote by u_Δ the induced control law.

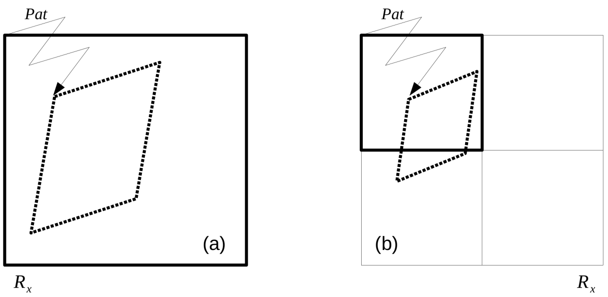

Fig. 2 Scheme of the bisection algorithm. The real boxes are hypercubes in n dimensions, they are represented here by rectangles.

The idea of the algorithm is the following: if we have an initial nodal vector $\{T\}$ belonging to a set $R_x = [T_{min}, T_{max}]^n$, we want to apply a pattern that keeps $\{T\}$ in R_x: the nodal temperatures, after application of the pattern, will still belong to $R_x = [T_{min}, T_{max}]^n$. In this purpose, we look for a pattern Pat that sends the whole set R_x in itself, such as in Figure 2(a). If we manage to do so, then, from any initial condition $\{T\}$ in R_x, we can apply Pat, and the nodal temperatures are sent in R_x, and we can thus reapply Pat infinitely. The temperature is stabilized in R_x. But as it is difficult to find a pattern that sends the whole set R_x in itself, we bisect R_x in sub-sets, and look for patterns which send the subsets in R_x, such as in Figure 2(b). We then have several patterns that send a partition of R_x in R_x. Furthermore, when looking for stabilizing patterns, we add the more restrictive constraint that the corresponding output of the images of the sets are sent in R_y, so that the output of the system reaches a smaller target set. The proofs of the efficiency of the decomposition algorithm and the control induced by the decomposition are given in [14,16,15].

Algorithms 1 and 2 show the main functions used by the state-space decomposition algorithm. At the beginning, the function "Bisection" calls sub-function "Find_Pattern" in order to get a k-pattern Pat such that $Post_{Pat}(R_x) \subseteq R_x$ and $Post_{Pat,C}(R_x) \subseteq R_y$. If it succeeds, then it is done. Otherwise, it divides R_x into 2^n sub-boxes $V_1, \dots, V_{2^n}$ of equal size. If for each V_i, Find_Pattern gets a k-pattern Pat_i such that $Post_{Pat_i}(V_i) \subseteq R_x$ and $Post_{Pat_i,C}(V_i) \subseteq R_y$, it is done. If, for some V_j, no such input pattern exists, the function is recursively applied to V_j. It ends with success when a successful decomposition of (R_x, R_y, k) is found, or failure when the maximal degree d of bisection is reached. The main function Bisection(W, R_x, R_y, D, K) is called with R_x as input value for W, d for input value for D, and k as input value for K; it returns either $\langle\{(V_i, Pat_i)\}_i, True\rangle$ with

$$\bigcup_i V_i = W,$$

$$\bigcup_i Post_{Pat_i}(V_i) \subseteq R_x,$$

$$\bigcup_i Post_{Pat_i,C}(V_i) \subseteq R_y$$

when it succeeds, or $\langle\_, False\rangle$ when it fails. Function Find_Pattern(W,R_x,R_y,K) looks for a K-pattern Pat for which $Post_{Pat}(W) \subseteq R_x$ and $Post_{Pat,C}(W) \subseteq R_y$: it selects all the K-patterns by increasing length order until either it finds such an input pattern Pat (output: $\langle Pat, True\rangle$), or none exists (output: $\langle\_, False\rangle$).

4 Model Order Reduction

The main drawback of the previous state-space decomposition algorithm is the computational cost, with a complexity in $O(2^{nd}N^k)$, with n the state-space dimension, d the maximum degree of decomposition, N the number of modes and k the maximum length of researched patterns. It is thus subject to the *curse of dimensionality.* In practice, the dimension n must be lower than 15 for acceptable computation times. Thus, by directly applying the bisection algorithm to a discretized PDE, the number of degrees of freedom is limited to 15 for a first order PDE, and even less for a

Algorithm 1: Bisection(W, R_x, R_y, D, K)

Input: A box W, a box R_x, a box R_y, a degree D of bisection, a length K of input pattern
Output: $\langle\{(V_i, Pat_i)\}_i, True\rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i) \subseteq R_y$, or $\langle\_, False\rangle$

$(Pat, b) := Find\_Pattern(W, R_x, R_y, K)$
if $b = True$ **then**
  return $\langle\{(W, Pat)\}, True\rangle$
else
  if $D = 0$ **then**
    return $\langle\_, False\rangle$
  else
    Divide equally W into $(W_1, \ldots, W_{2^n})$
    for $i = 1 \ldots 2^n$ **do**
      $(\Delta_i, b_i) :=$ Bisection(W_i,R_x,R_y,$D-1$,K)
    return $(\bigcup_{i=1\ldots 2^n} \Delta_i, \bigwedge_{i=1\ldots 2^n} b_i)$

Algorithm 2: Find_Pattern(W, R_x, R_y, K)

Input: A box W, a box R_x, a box R_y, a length K of input pattern
Output: $\langle Pat, True\rangle$ with ,$Post_{Pat}(W) \subseteq R_x$,$Post_{Pat,C}(W) \subseteq R_y$ and $Unf_{Pat}(W) \subseteq S$, or $\langle\_, False\rangle$ when no input pattern maps W into R_x and CW into R_y

for $i = 1 \ldots K$ **do**
  $\Pi :=$ set of input patterns of length i
  while Π *is non empty* **do**
    Select Pat in Π
    $\Pi := \Pi \setminus \{Pat\}$
    if $Post_{Pat}(W) \subseteq R_x$ *and* $Post_{Pat,C}(W) \subseteq R_y$ **then**
      return $\langle Pat, True\rangle$
return $\langle\_, False\rangle$

higher order PDE written in state-space representation. The use of a Model Order Reduction (MOR) is thus unavoidable.

We choose here to use *projection-based* model order reduction methods [4]. Given a full-order system Σ, an interest set $R_x \subset \mathbb{R}^n$ and an objective set $R_y \subset \mathbb{R}^m$, we construct a reduced-order system $\hat{\Sigma}$ using a projection π of $\mathbb{R}^n$ to $\mathbb{R}^{n_r}$. If $\pi \in \mathbb{R}^{n\times n}$ is a projection, it verifies $\pi^2 = \pi$, and π can be written as $\pi = \pi_L \pi_R$, where $\pi_L \in \mathbb{R}^{n\times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$ and $n_r = rank(\pi)$. The reduced-order system $\hat{\sigma}$ is then obtained by the change of variable $\hat{x} = \pi_R x$:

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

with

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

The projection π can be constructed by multiple methods: Proper Orthogonal Decomposition [12,25], balanced truncation [7,3,29,8], balanced POD [39]... We use here the balanced truncation method, widely used in the control community and particularly adapted to the models used here, written under state-space representation.

The objective is now to compute a decomposition at the low order level, and apply the induced reduced control to the full order system. In order to ensure that the reduced control is effective, we introduce the following notations:

- $\hat{\mathbf{x}}(t, \hat{x}, u)$ denotes the point reached by $\hat{\Sigma}$ at time t under mode $u \in U$ from the initial condition $\hat{x}$.
- $\mathbf{y}(t, x, u)$ denotes the output point reached by Σ at time t under mode $u \in U$ from the initial condition x.
- $\mathbf{y_r}(t, \hat{x}, u)$ denotes the output point reached by $\hat{\Sigma}$ at time t under mode $u \in U$ from the initial condition $\hat{x}$.

When a control u is applied to both full-order and reduced-order systems, an error between the output trajectories $\mathbf{y}(t, x, u)$ and $\mathbf{y_r}(t, \pi_R x, u)$ is unavoidable, and we denote it by $e_y(t, x, u)$. A first tool to ensure the effectiveness of the reduced-order control is to compute a bound on $\|e_y(t, x, u)\|$. A second source of error is the deviation between $\pi_R \mathbf{x}(t, x, u)$ and $\hat{\mathbf{x}}(t, \pi_R x, u)$, which we denote by $e_x(t, x, u)$. Computing a bound on $\|e_x(t, x, u)\|$ will also be necessary. Before establishing these error bounds, we first briefly describe the balanced truncation method. We then present how we compute a reduced-order control and apply it to the full-order system.

4.1 The Balanced Truncation

Applying the balanced truncation consists in balancing then truncating the system. Balancing the system requires finding balancing transformations which diagonalize the controllability and observability gramians of the system in the same basis.

The controllability and observability gramians W_c and W_o of the system Σ are respectively the solutions of the dual (infinite-time horizon) Lyapunov equations

$$AW_c + W_c A^\top + BB^\top = 0 \tag{4}$$

and

$$A^\top W_o + W_o A + C^\top C = 0 \tag{5}$$

The balancing transformations π_R and π_L are then computed as follows [8]:

1. Compute the Cholesky factorization $W_c = UU^\top$

2. Compute the eigenvalue decomposition of $U^\top W_o U$
$U^\top W_o U = K\sigma^2 K^\top$
where the entries in σ are ordered by decreasing order
3. Compute the transformations
$\pi_R = \sigma^{-\frac{1}{2}} K^\top U^{-1}$
$\pi_L = UK\sigma^{-\frac{1}{2}}$
One can then verify that

$$\pi_R W_c \pi_R^\top = \pi_L^\top W_o \pi_L = \sigma$$

and σ contains the Hankel singular values of the system.

Computing the balancing transformations for large scale systems derived for example from discretized partial differential equations are usually very expensive - even sometimes irrelevant - and many advances have been carried out in order to solve the Lyapunov equations and compute the transformations with approximate methods, often based on Krylov subspace methods (see for example [3,31,7]).

4.2 Error Bounding

4.2.1 Error bounding for the output trajectory

Here, a scalar *a posteriori* error bound for e_y is given (mainly inspired from [21]). The error bound ε_y can be computed from simulations of the full and reduced-order systems. The computation time for simulations is negligible compared with that of the bisection method to generate the decompositions.

Computing an upper bound of $\|e_y(t,x,u)\|$ is equivalent to seeking the solution of the following (optimal control) problem:

$$\begin{aligned}\varepsilon_y(t) &= \sup_{u\in U, x_0\in R_x} \|e(t,x_0,u)\| \\ &= \sup_{u\in U, x_0\in R_x} \|\mathbf{y}(t,x_0,u) - \mathbf{y_r}(t,\pi_R x_0,u)\|.\end{aligned}$$

Since the full-order and reduced-order systems are linear, one can use a superposition principle and the error bound can be estimated as $\varepsilon_y(t) \le \varepsilon^{x_0=0}(t) + \varepsilon^{u=0}(t)$ where $\varepsilon_y^{x_0=0}$ is the error of the zero-state response, given by (see [21])

$$\begin{aligned}\varepsilon_y^{x_0=0}(t) &= \max_{u\in U} \|u\| \cdot \|e_y(t,x_0=0,u)\| \\ &= \max_{u\in U} \|u\| \cdot \|\mathbf{y}(t,0,u) - \mathbf{y_r}(t,0,u)\|,\end{aligned}$$

and $\varepsilon_y^{u=0}$ is the error of the zero-input response, given by

$$\begin{aligned}\varepsilon_y^{u=0}(t) &= \sup_{x_0\in R_x} \|e_y(t,x_0,u=0)\| \\ &= \sup_{x\in R_x} \|\mathbf{y}(t,x_0,0) - \mathbf{y_r}(t,\pi_R x_0,0)\|.\end{aligned}$$

Using some algebraic manipulations (see [21]), one can find a precise bound for $\varepsilon_y^{x_0=0}$ and $\varepsilon_y^{u=0}$:

$$\varepsilon_y^{x_0=0}(t) \le \|u(\cdot)\|_\infty^{[0,t]} \int_0^t \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt, \quad (6)$$

$$\varepsilon_y^{u=0}(t) \le \sup_{x_0\in R_x} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|. \quad (7)$$

The first error bound (6) always increases with time whereas the second bound (7) can either increase or decrease. These properties are used to compute a guaranteed bound. For all $j \in \mathbb{N}$ (j corresponds to the length of the pattern applied), we have:

$$\varepsilon_y(j\tau) \le \varepsilon_y^j$$

with

$$\begin{aligned}\varepsilon_y^j = \|u(\cdot)\|_\infty^{[0,j\tau]} &\int_0^{j\tau} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt \\ &+ \sup_{x_0\in R_x} \| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau\hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|. \quad (8)\end{aligned}$$

Furthermore, we have:

$$\forall t \ge 0, \quad \varepsilon_y(t) \le \varepsilon_y^\infty$$

with

$$\varepsilon_y^\infty = \sup_{t\ge 0} \varepsilon_y(t). \quad (9)$$

This bound exists when the modulus of the eigenvalues of $e^{\tau A}$ and $e^{\tau\hat{A}}$ is strictly inferior to one, which we suppose here.

4.2.2 Error bounding for the state trajectory

Denoting by $j \in \mathbb{N}$ the length of the pattern applied, the following results holds:

$$\mathbf{x}(t=j\tau,x,u) = e^{j\tau A}x + \int_0^{j\tau} e^{A(j\tau-t)}Bu(t)dt,$$

$$\hat{\mathbf{x}}(t=j\tau,\pi_R x,u) = e^{j\tau\hat{A}}\pi_R x + \int_0^{j\tau} e^{\hat{A}(j\tau-t)}\hat{B}u(t)dt,$$

Using an approach similar to the construction of the bounds (6) and (7), we obtain the following bound, which depends on the length j of the pattern applied:

$$\|\pi_R\mathbf{x}(t=j\tau,x,u) - \hat{\mathbf{x}}(t=j\tau,\pi_R x,u)\| \le \varepsilon_x^j, \quad (10)$$

with

$$\begin{aligned}\varepsilon_x^j = \|u(\cdot)\|_\infty^{[0,j\tau]} &\int_0^{j\tau} \| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \| dt \\ &+ \sup_{x_0\in R_x} \| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau\hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \|. \quad (11)\end{aligned}$$

Remark: in order to simplify the reading, the notation $|Pat|$ will often be used in the following to denote the length of the pattern Pat.

5 Reduced Order Control

Two procedures are proposed for synthesizing reduced-order controllers: (i) an offline procedure, consisting in computing a complete sequence of control inputs for a given initial condition; (ii) an online procedure, where the patterns are computed through online projection of the full-order state. We describe these approaches in the following subsections.

5.1 Offline Procedure

Suppose that we are given a system Σ, an interest set R_x, and an objective set R_y. The reduced-order system $\hat{\Sigma}$ of order n_r, obtained by balanced truncation, is written under the form of equation (3):

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

where $\hat{A} = \pi_R A \pi_L \in \mathbb{R}^{n_r \times n_r}$, $\hat{B} = \pi_R B \in \mathbb{R}^{n_r \times p}$, $\hat{C} = C\pi_L \in \mathbb{R}^{m \times n_r}$.

We denote by $\hat{R}_x$ the projection of R_x. Given the interest set $\hat{R}_x$, the objective set R_y and a maximal length of researched pattern K, the application of the state-space decomposition algorithm to the reduced system returns, when it succeeds, a decomposition $\hat{\Delta}$ of the form $\{\hat{V}_i, Pat_i\}_{i\in I}$, with I a finite set of indices, such that:

1. $\bigcup_{i\in I} \hat{V}_i = \hat{R}_x$,
2. for all $i \in I$: $Post_{Pat_i}(\hat{V}_i) \subseteq \hat{R}_x$,
3. for all $i \in I$: $Post_{Pat_i,\hat{C}}(\hat{V}_i) \subseteq R_y$.

The decomposition $\hat{\Delta}$ induces a control $u_{\hat{\Delta}}$ on $\hat{R}_x$. Applied on the reduced-order system $\hat{\Sigma}$, the control $u_{\hat{\Delta}}$ keeps $\hat{x}$ in $\hat{R}_x$ and sends y_r in R_y. This control can be applied to the full-order system in two steps: a sequence of patterns is computed on the reduced-order system, and it is then applied to the full order system:

(a) Let x_0 be an initial condition in R_x. Let $\hat{x}_0 = \pi_R x_0$ be its projection belonging to $\hat{R}_x$, $\hat{x}_0 = \pi_R x_0$ is the initial condition for the reduced system $\hat{\Sigma}$: $\hat{x}_0$ belongs to $\hat{V}_{i_0}$ for some $i_0 \in I$; thus, after applying Pat_{i_0}, the system is led to a state $\hat{x}_1$; $\hat{x}_1$ belongs to $\hat{V}_{i_1}$ for some $i_1 \in I$; and iteratively, we build, from an initial state $\hat{x}_0$, a sequence of states $\hat{x}_1, \hat{x}_2, \ldots$ obtained by application of the sequence of k-patterns Pat_{i_0}, Pat_{i_1}, ... (steps (1), (2) and (3) of Figure 3).

(b) The sequence of k-patterns is computed for the reduced system $\hat{\Sigma}$, but it can be applied to the full-order system Σ: we build, from an initial point x_0, a sequence of points x_1, x_2,... by application of the k-patterns Pat_{i_0},Pat_{i_1},... (steps (4), (5) and (6) of Figure 3). Moreover, for all $x_0 \in R_x$ and for all $t \geq 0$, the error $\|\mathbf{y}(t, x_0, u) - \mathbf{y_r}(t, \pi_R x_0, u)\|$ is bounded by ε_y^∞, as defined in equation(9).

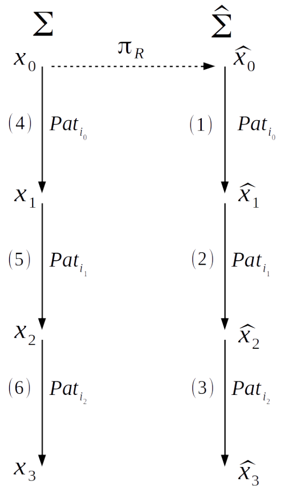

Fig. 3 Diagram of the offline procedure for a simulation of length 3.

This procedure thus allows, for any system Σ of the form (3), and given an interest set R_x and an objective set R_y, to send the output of the full-order system in the set $R_y + \varepsilon_y^\infty$. More precisely, if $\hat{\Sigma}$ is the projection by balanced truncation of Σ, let $\hat{\Delta}$ be a decomposition for ($\hat{R}_x$,R_y,k) w.r.t. $\hat{\Sigma}$. Then, for all $x_0 \in R_x$, the induced control $u_{\hat{\Delta}}$ applied to the full-order system Σ in x_0 is such that for all $j > 0$, the output of the full-order system $y(t)$ returns to $R_y + \varepsilon_y^\infty$ after at most k τ-steps.

Here, $R_y + \varepsilon_y^\infty$ denotes the set containing R_y with a margin of ε_y^∞. If R_y is an interval product of the form $[a_1, b_1] \times \cdots \times [a_m, b_m]$, then $R_y + \varepsilon_y^\infty$ is defined by $[a_1 - \varepsilon_y^\infty, b_1 + \varepsilon_y^\infty] \times \cdots \times [a_m - \varepsilon_y^\infty, b_m + \varepsilon_y^\infty]$.

Remark: Here, we ensure that $\mathbf{y}(t, x_0, u)$ is in $R_y + \varepsilon_y^\infty$ at the end of every pattern, but an easy improvement is to ensure that $\mathbf{y}(t, x_0, u)$ stays in a safety set $S_y \supset R_y$ at every step of time $k\tau$. Indeed, as explained in [14], we can ensure that the unfolding of the output trajectory stays in a given safety set S_y. The unfolding of the output of a set is defined as follows: given a pattern Pat of the form $(u_1 \cdots u_m)$, and a set $X \subset \mathbb{R}^n$, the *unfolding of the output of* X *via* Pat, denoted by $Unf_{Pat,C}(X)$, is the set $\bigcup_{i=0}^{m} X_i$ with:

– $X_0 = \{Cx | x \in X\}$,
– $X_{i+1} = Post_{u_{i+1},C}(X_i)$, for all $0 \leq i \leq m-1$.

The unfolding thus corresponds to the set of all the intermediate outputs produced when applying pattern

Pat to the states of X. In order to guarantee that $\mathbf{y}(t, x_0, u)$ stays in S_y, we just have to make sure that $\mathbf{y_r}(t, \pi_R x_0, u)$ stays in the reduced safety set $S_y - \varepsilon_y^\infty$. We thus have to add, in the line 6 of Algorithm 2, the condition: "and $Unf_{Pat,C}(W) \subset S_y - \varepsilon_y^\infty$".

5.2 Online Procedure

Up to this point, the procedure of control synthesis consists in computing a complete sequence of patterns on the reduced order model $\hat{\Sigma}$ for a given initial state x_0, and applying the pattern sequence to the full-order model Σ. The entire control law is thus computed offline. While the decomposition is always performed offline, one can however use the decomposition $\hat{\Delta}$ online as follows: let x_0 be the initial state in R_x and $\hat{x}_0 = \pi_R x_0$ (step (1) of Figure 4) its projection belonging to $\hat{R}_x$, $\hat{x}_0$ belongs to $\hat{V}_{i_0}$ for some $i_0 \in I$; we can thus apply the associated pattern Pat_{i_0} to the full-order system Σ, which yields a state $x_1 = \mathbf{x}(|Pat_{i_0}|\tau, x_0, Pat_{i_0})$ (step (2) of Figure 4), the corresponding output is sent to $y_1 = \mathbf{y}(|Pat_{i_0}|\tau, x_0, Pat_{i_0}) \in R_y + \varepsilon_y^{|Pat_{i_0}|}$; in order to continue to step (3), we have to guarantee that $\pi_R \mathbf{x}(|Pat_i|\tau, x, Pat_i))$ belongs to $\hat{R}_x$ for all $x \in R_x$ and for all $i \in I$. As explained below, this is possible using the computation of an upper bound to the error $\|\pi_R \mathbf{x}(|Pat_i|\tau, x, Pat_i) - \hat{\mathbf{x}}(|Pat_i|\tau, \pi_R x, Pat_i)\|$ and a reinforcement of the procedure for taking into account this error.

Let $\varepsilon_x^{|Pat|}$ be the upper bound to

$$\|\pi_R \mathbf{x}(|Pat|\tau, x, Pat) - \hat{\mathbf{x}}(|Pat|\tau, \pi_R x, Pat)\|,$$

as defined in equation (11). We modify the Algorithms 1 and 2, which become "Bisection_Dyn" and "Find_Pattern_Dyn" (Algorithms 3 and 4), they are computed with an additional input $\varepsilon_x = (\varepsilon_x^1, \ldots, \varepsilon_x^k)$, k being the maximal length of the patterns. With such an additional input, we perform an *ε-decomposition.* Given a system Σ, two sets R_x and R_y respectively subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, a positive integer k, and a vector of errors $\varepsilon_x = (\varepsilon_x^1, \ldots, \varepsilon_x^k)$, application of the ε-decomposition returns a set Δ of the form $\{V_i, Pat_i\}_{i \in I}$, where I is a finite set of indexes, every V_i is a subset of R_x, and every Pat_i is a k-pattern such that:

(a') $\bigcup_{i \in I} V_i = R_x$,
(b') for all $i \in I$: $Post_{Pat_i}(V_i) \subseteq R_x - \varepsilon_x^{|Pat_i|}$,
(c') for all $i \in I$: $Post_{Pat_i,C}(V_i) \subseteq R_y$.

Note that condition (b') is a strengthening of condition (b) in subsection 3.2. Accordingly, line 6 of Algorithm 2 becomes in Algorithm 4:

6 **if** $Post_{Pat}(W) \subseteq R_x - \varepsilon_x^i$ *and* $Post_{Pat,C}(W) \subseteq R_y$ **then**

The new algorithms enable to guarantee that the projection of the full-order system state $\pi_R x$ always stays in $\hat{R}_x$, we can thus perform the online control as follows:

Since $Post_{Pat_{i_0}}(\hat{V}_{i_0}) \subseteq \hat{R}_x - \varepsilon_x^{|Pat_{i_0}|}$ and $\pi_R x_0 \in \hat{V}_{i_0}$, we have $Post_{Pat_{i_0}}(\pi_R x_0) \in \hat{R}_x - \varepsilon_x^{|Pat_{i_0}|}$; thus $\pi_R x_1 = \pi_R \mathbf{x}(|Pat_{i_0}|\tau, x_0, Pat_{i_0})$ belongs to $\hat{R}_x$, because $\varepsilon_x^{|Pat_{i_0}|}$ is a bound of the maximal distance between $\hat{\mathbf{x}}(|Pat_{i_0}|\tau, \pi_R x_0, Pat_{i_0})$ and $\pi_R \mathbf{x}(|Pat_{i_0}|\tau, x_0, Pat_{i_0})$; since $\pi_R x_1$ belongs to $\hat{R}_x$, it belongs to V_{i_1} for some $i_1 \in I$; we can thus compute the input pattern Pat_{i_1}, and therefore, we can reapply the procedure and compute an input pattern sequence Pat_{i_0},Pat_{i_1},... As for the output, the yielded points $y_1 = \mathbf{y}(|Pat_{i_0}|\tau, x_0, Pat_{i_0})$, $y_2 = \mathbf{y}(|Pat_{i_1}|\tau, x_1, Pat_{i_1})$, ... belong respectively to the sets $R_y + \varepsilon_y^{|Pat_{i_0}|}$,$R_y + \varepsilon_y^{|Pat_{i_1}|}$,...

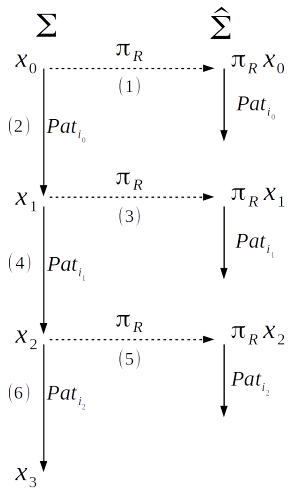

Fig. 4 Diagram of the online procedure for a simulation of length 3.

The main advantage of such an online control is that the estimated errors $\varepsilon_y^{|Pat_{i_0}|}$,$\varepsilon_y^{|Pat_{i_1}|}$,... are dynamically computed, and are smaller than the static bound ε_y^∞ used in the offline control. The price to be paid is the strengthening of condition (b'). In the best case, i.e. if the errors are low and the system is very contractive, this can result in the same decomposition and computation time as in the offline procedure. But if the system is not contractive enough or if the errors are too large, this can lead to a more complicated decomposition, and thus higher computation times, and in the worst case, no successful decomposition at all.

Algorithm 3: Bisection_Dyn$(W, R_x, R_y, D, K, \varepsilon_x)$

Input: A box W, a box R_x, a box R_y, a length K of pattern, a vector of errors ε_x, a degree D of bisection

Output: $\langle\{(V_i, Pat_i)\}_i, True\rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i) \subseteq R_y$, or $\langle\_, False\rangle$

(Pat, b) :=Find_Pattern_Online$(W, R_x, R_y, K, \varepsilon_x)$
if $b = True$ **then**
  return $\langle\{(W, Pat)\}, True\rangle$
else
  if $D = 0$ **then**
    return $\langle\_, False\rangle$
  else
    Divide equally W into $(W_1, \ldots, W_{2^n})$
    for $i = 1 \ldots 2^n$ **do**
      (Δ_i, b_i) := Bisection_Online$(W_i, R_x, R_y, K, \varepsilon_x, D-1)$
    return $(\bigcup_{i=1\ldots 2^n} \Delta_i, \bigwedge_{i=1\ldots 2^n} b_i)$

Algorithm 4: Find_Pattern_Dyn$(W, R_x, R_y, K, \varepsilon_x)$

Input: A box W, a box R_x, a box R_y, a length K of pattern, a vector of errors ε_x

Output: $\langle Pat, True\rangle$ with $,Post_{Pat}(W) \subseteq R_x, Post_{Pat,C}(W) \subseteq R_y$ and $Unf_{Pat}(W) \subseteq S$, or $\langle\_, False\rangle$ when no pattern maps W into R_x and CW into R_y

for $i = 1 \ldots K$ **do**
  Π := set of patterns of length i
  while Π *is non empty* **do**
    Select Pat in Π
    $\Pi := \Pi \setminus \{Pat\}$
    if $Post_{Pat}(W) \subseteq R_x - \varepsilon_x^i$ *and* $Post_{Pat,C}(W) \subseteq R_y$ **then**
      return $\langle Pat, True\rangle$
return $\langle\_, False\rangle$

6 Numerical Results

6.1 Thermal Problem on a Metal Plate

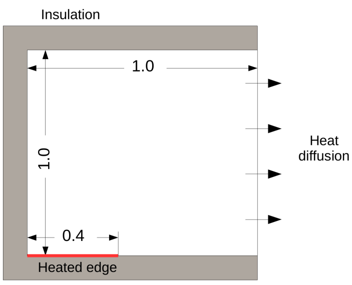

Fig. 5 Geometry of the square plate.

We consider here the problem of controlling the central node temperature of a square metal plate, discretized by finite elements; this example is taken from [22]. The square plate is subject to the heat equation: $\frac{\partial T}{\partial t}(x,t) - \alpha \Delta T(x,t) = 0$. After discretization, the system is written under its state-space representation (3). The plate is insulated along three edges, while the right edge is open. The left half of the bottom edge is connected to a heat source. The exterior temperature is set to 0°C, the temperature of the heat source is either 0°C (mode 0) or 1°C (mode 1). The heat transfers with the exterior and the heat source are modeled by a convective transfer. The full-order system state corresponds to the nodal temperatures. The output is the temperature of the central node. The system is reduced from $n = 897$ to $n_r = 2$ (Figure 7) and $n_r = 3$ (Figure 8). The interest set is $R_x = [0, 0.15]^{897}$ and the objective set $R_y = [0.06, 0.09]$. The sampling time is set to $\tau = 8$ s. The geometry of the system is given in Figure 5. The decomposition obtained with the offline procedure is given in Figure 6.

The decompositions and simulations have been performed with MINIMATOR (an Octave code available at https://bitbucket.org/alecoent/minimator_red) on a 2.80 GHz Intel Core i7-4810MQ CPU with 8 GB of memory. The decompositions were obtained in 5 seconds for the case $n_r = 2$ and in 2 minutes for the case $n_r = 3$.

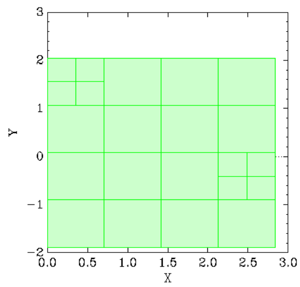

Fig. 6 Decomposition of $\hat{R}_x = \pi_R R_x$ in the plane $(\hat{x}_1, \hat{x}_2)$ (for $n_r = 2$) with the offline procedure.

Simulations of the offline and online methods are given in Figures 7 and 8. We notice in Figure 7 that the trajectory y (resp. y_r) exceeds the objective set R_y (resp. $R_y + \varepsilon_y^{|Pat_i|}$) during the application of the second pattern, yet the markers corresponding to the end of input patterns do belong to objective sets. Comparing the cases $n_r = 2$ and $n_r = 3$, we finally observe that a less reduced model causes lower error bounds, and thus a more precise control, at the expense of a higher computation time.

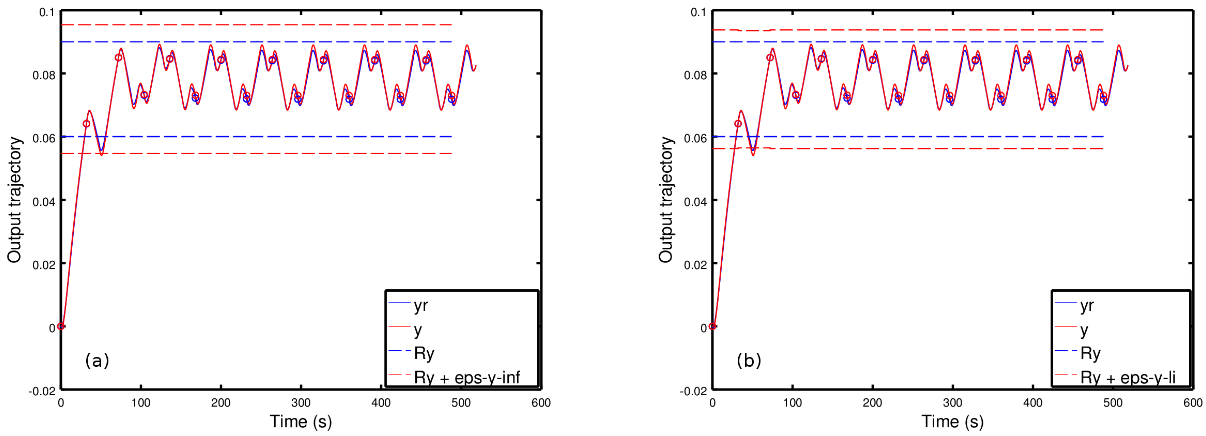

Fig. 7 For $n_r = 2$, simulation of $y(t) = Cx(t)$ and $y_r(t) = \hat{C}\hat{x}(t)$ from the initial condition $x_0 = (0)^{897}$. (a): guaranteed offline control; (b): guaranteed online control.

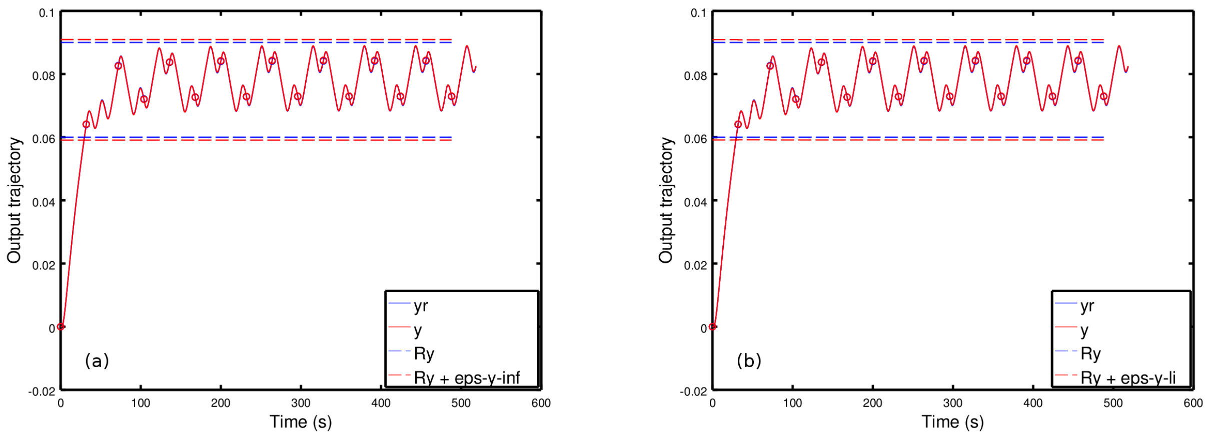

Fig. 8 For $n_r = 3$, simulation of $y(t) = Cx(t)$ and $y_r(t) = \hat{C}\hat{x}(t)$ from the initial condition $x_0 = (0)^{897}$. (a): guaranteed offline control; (b): guaranteed online control.

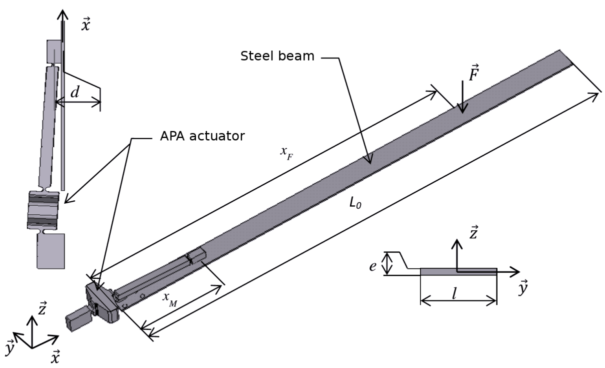

Fig. 9 Scheme of the vibrating beam.

6.2 Vibrating Beam

In this case study, which comes from a practical work designed by Fabien Formosa [13], we apply our method to vibration control of a cantilever beam. The objective is to keep the tip displacement of the beam as close as possible to zero. To stabilize the beam, a piezoelectric patch applies a torque with the mechanism schemed in Figure 9 at a distance x_M from the blocked side of the beam. The model retained is a finite element model with classical beam elements. The beam equation is the following:

$$m\ddot{w}(x,t) + EI\frac{\partial^4 w(x,t)}{\partial x^4} = \frac{\partial M_u}{\partial x}\delta(x - x_M) \tag{12}$$

The torque M_u is chosen with the control variable u. By applying the right torque at the right time, we hope to stabilize the beam. In its finite element writing, the system is:

$$M\ddot{W} + KW = F_u \tag{13}$$

Using a modal decomposition

$$W(x,t) = \sum_{i \le n_{modes}} a_i(t)\varphi_i(x),$$

we can write a reduced system of the form:

$$M_r\ddot{a}_i(t) + 2\zeta_i\dot{a}_i(t) + K_r a_i(t) = F_{r,u}. \quad (14)$$

Note that a modal damping is added in this step, it permits to have a realistic behaviour of the beam since it is subject to loss of energy. By rearranging the terms of equation (14) into a first order ODE, we can write the system under a state-space representation:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \quad (15)$$

where the output y is the tip displacement of the beam. Henceforth, the state variable contains the variables a_i and $\dot{a}_i$. The dimension of the state-space is thus twice the number of retained modes. In this way, the system can be treated with the method developed here, applying a balanced truncation to the system (15) and building a reduced-order control.

Note that the intermediate model order reduction by modal decomposition cannot actually be avoided, because the direct rearrangement of system (13) into its state-space representation leads to a matrix A possessing some positive eigenvalues (instead of only negative ones), and the calculation of balancing transformations is then much more complicated, or even impossible.

The finite element model is composed of 60 elements (thus 120 degrees of freedom to take the rotation into account), we retain 20 modes for the modal decomposition, and the system is reduced to $n_r = 4$. Nine control modes are chosen to control the beam, including the mode corresponding to a null torque. Two simulations for different initial conditions and objective sets are given in Figure 10. In the first one, several modes are initially excited, whereas only the first mode is excited in the second one. In both cases, the online procedure is applied, and we manage to stabilize the tip displacement relatively fast. The output of the full-order system is stabilized in $R_y + \varepsilon_y^{|Pat_i|}$ with $\varepsilon_y^{|Pat_i|} \backsimeq 0.2$. The errors $\varepsilon_y^{|Pat_i|}$ can seem quite high compared to the tip displacement, this comes from the hyperbolic nature of the equations which rule this example. However, in a practical point of view, this is clear that the reduced-order output fits well the behavior of the full-order system.

6.3 Vibrating Aircraft Panel

In order to verify the handling of higher dimensional systems, we apply our method to the vibration control of an aircraft panel. This example, taken from [24], consists in stabilizing the panel as close as possible to the equilibrium, which corresponds to a null displacement inside the whole panel. In this purpose, seven piezoelectric patches are glued on the panel, one is used for exciting the panel (patch 1 of Figure 11), one is used as a sensor to evaluate the performance of the control (patch 2), one is used for the observation of modal states (patch 6), and three are used for vibration control (patches 3 to 5), the last patch being used to validate the reconstruction (patch 7). For the numerical simulations, we choose the measurements of the sensor patch as the output of the system.

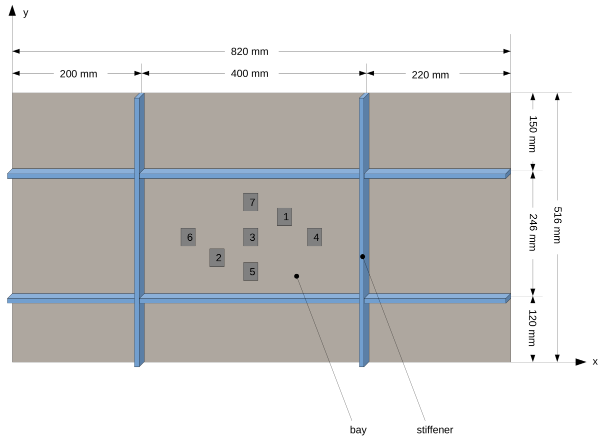

Fig. 11 Scheme of the vibrating aircraft panel.

Just as the cantilever beam, we use a finite element model reduced by modal decomposition then balanced truncation. The system is written exactly in the same way, but with shell elements, and thus six degrees of freedom by node. The finite shell element model consists of 57000 degrees of freedom. We retain 50 modes for the modal decomposition, and the model is reduced down to $n_r = 5$ by balanced truncation. Seven control modes are used for vibration control, it corresponds to a null voltage applied on all the control patches, a positive constant voltage applied on each control patch (one patch is subject to a voltage at a time), and a negative constant voltage applied on each control patch. The reader is referred to [24] for more information on the exact functioning of the piezoelectric patches used in this case study, and see for example [20,28] for more general information on piezoelectric patches and their use for structural damping. With the same hardware configuration as in the previous example, the computation of a decomposition took nearly a week. A simulation of the online procedure is given in Figure 12 and 13.

We observe that the response of the controlled full-order system is better than the non-controlled one, the main peaks observed in the non-controlled response are avoided. Nevertheless, the stabilization is not as efficient as one may expect. One can see that the reduced-order system is however well stabilized. This points out that the model reduction does not catch, in this case, all the information needed for control purposes. While we

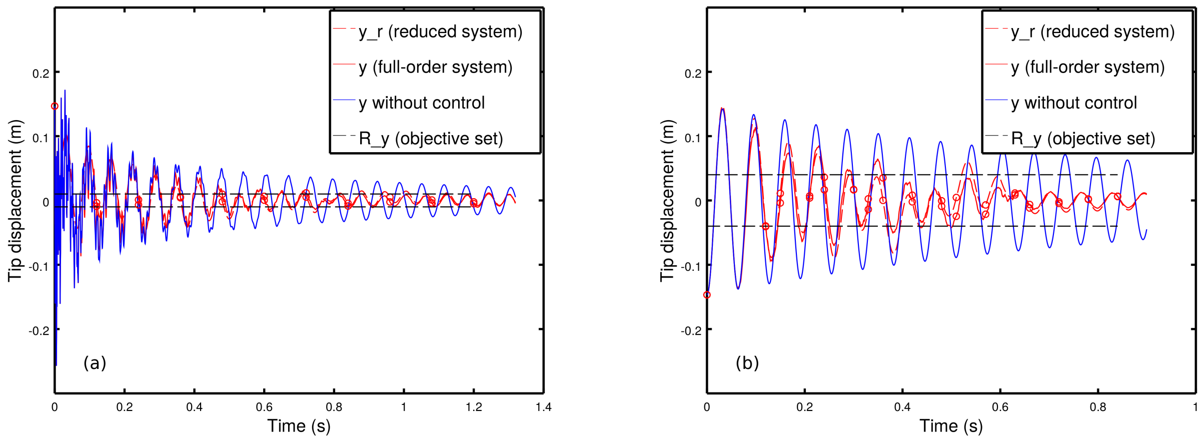

Fig. 10 Simulations of vibration control of the cantilever beam for two different initial conditions and objective boxes. (a): several modes excited; (b): first mode excited.

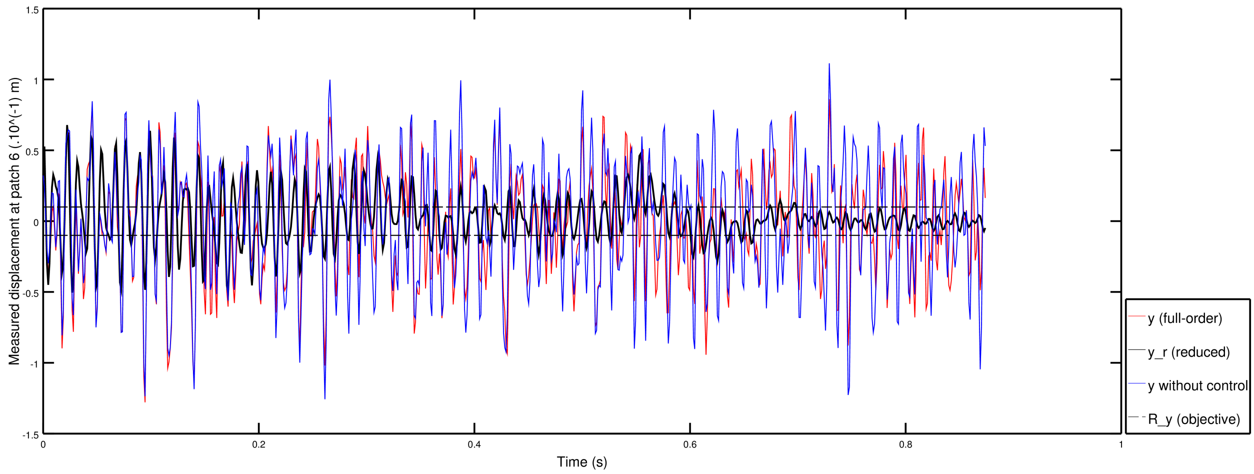

Fig. 12 Simulation of vibration control of the aircraft panel.

are currently investigating new model reduction techniques, adapted to hyperbolic and non-linear systems, we also think that in practice, the stabilization would be better because of the smoothness appearing in the applied torques in a real application.

7 Extension to Output Feedback Control

So far, we designed reduced state-dependent controllers for switched control systems, permitting to stabilize the output of the system in a given objective set R_y. During a real online use, one is only supposed to know a part of the state of the system, such as measurements of sensors. We now want to take these partial measurements into account, by adding an intermediate step in the online use, namely, observation. We suppose that only the output of the system is known online. In the next sub-section, we introduce the principle of observation and give some preliminary results justifying the use of observers for switched control systems, allowing us to adapt our algorithms to the use of observers. We then present some numerical results of the use of observers with model order reduction. The whole approach with model order reduction is schemed in Figure 14, but as we do not have any proof for the efficiency of the use of observers with model order reduction, we only provide some numerical simulations. We are currently working on the establishment an error bound taking into account the projection error and the observation error, that will permit to construct a guaranteed reduced observer based control.

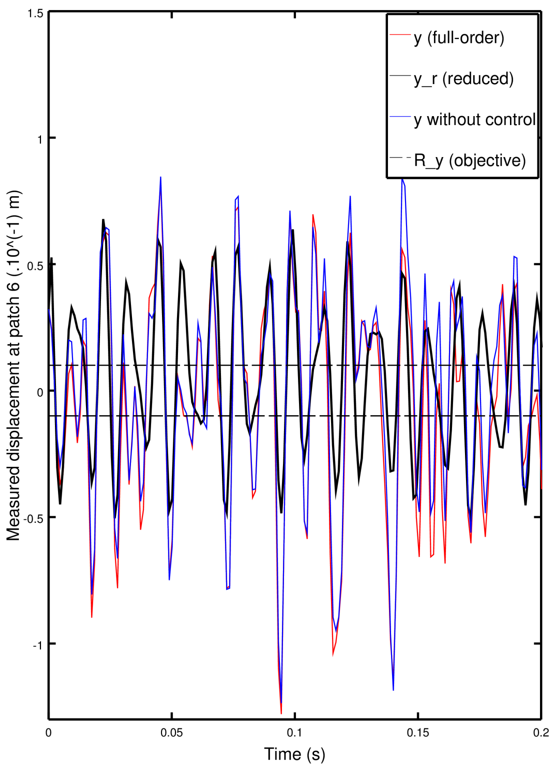

Fig. 13 Enlargement of Figure 13 on the time interval $[0, 0.2]$.

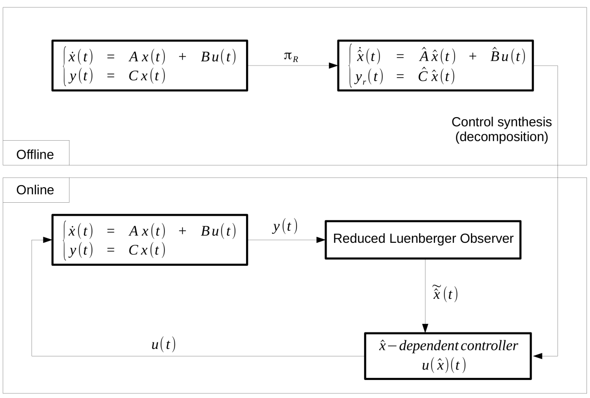

Fig. 14 Principle of the output feedback control

7.1 Partial observation

Having defined the state-space bisection algorithm for switched control systems with output, we now add the constraint that the system is partially observed. The objective is to design an *output feedback* controller using the state-space bisection algorithm introduced above.

We recall that the switched system Σ is written under the following form:

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

We suppose that during an online use, one is only supposed to know $y(t)$ (we suppose that y can be measured in real time, that is at every time t). If just this partial information of the state is known, we cannot directly apply our state-dependent controller synthesis method. An intermediate step must be introduced: the reconstruction of the state. The reconstruction is made with the help of an observer: it is an intermediate system that provides an estimate of the state of the system Σ from the measurements of the output y and the input u of the system Σ. In fact, this means that we want to design an output feedback law for the system Σ with the help of an observer. In this paper, we retain the Luenberger observer [40,2,1] to reconstruct the state of Σ, it is subject to the following equation:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n\times m} \tag{16}$$

Obviously, the observer does not reconstruct exactly the state x of the system Σ, we thus introduce the reconstruction error $\eta(t) = \|x(t) - \tilde{x}(t)\|$. Our goal is to control the system Σ with this estimate $\tilde{x}$: we apply a law $u(\tilde{x})$. One can note that the method relies on the convergence of the observer $\tilde{x}$ to the state x, this aspect is developed in the following section.

The entries of the control problem we retain are then the following:

- an interest set $R_x \subset \mathbb{R}^n$,
- an objective set $R_y \subset \mathbb{R}^m$,
- an initial, a priori known, reconstruction error η_0.

With the method given below, the outputs of the problem are the following:

- a decomposition of R_x w.r.t. η_0 and the dynamics of Σ,
- a procedure to choose u knowing $\tilde{x}$,
- and the guarantee that, for any pattern Pat, if $x_0 \in R_x$ and $\eta(0) \leq \eta_0$, then $\mathbf{x}(|Pat|\tau, x_0, Pat) \in R_x$ and $\mathbf{y}(|Pat|\tau, x_0, Pat) \in R_y$.

Let us now introduce some hypotheses and important results to ensure the efficiency of the method.

7.2 Convergence of the observer

The properties of the Luenberger observer depend on the choice of the matrices $L(u)$ appearing in (16). A crucial assumption in what follows is that it is possible to choose $L(\cdot)$ in such a way that the modes of the Luenberger observer share a common non-strict quadratic Lyapunov functions, i.e., there exists a positive definite matrix P such that:

$$\forall u, \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0. \tag{17}$$

The dynamics of the original switched system and of the Luenberger switch observer can be grouped in the augmented system

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} A - L(u)C & L(u)C \\ 0 & A \end{pmatrix} \begin{pmatrix} \tilde{x} \\ x \end{pmatrix} + \begin{pmatrix} Bu \\ Bu \end{pmatrix}.$$

Define $e(t) = x(t) - \tilde{x}(t)$ and $\eta(t) = e(t)^T P e(t)$. By definition $e(\cdot)$ satisfies

$$\dot{e} = (A - L(u)C)e \tag{18}$$

and assumption (17) implies that η is non-increasing along all trajectories. The patterns in $u(\cdot)$ will be chosen in order to guarantee that not only η decreases, but actually converges to zero.

An assumption which may be motivated by the technical constraints of the system under consideration is the existence of a *dwell-time*, that is, a positive constant τ such that two subsequent discontinuities of $u(\cdot)$ have a distance of at least τ (recall that $u(\cdot)$ is assumed to be piecewise constant). The dwell-time condition not only reflects technological constraints, but is also useful in the asymptotic analysis of the switched system (3). The basic result that we will use is a simplified version of [36, Theorem II.5], which states that under the dwell-time hypothesis, and by choosing properly the patterns, one can manage to make $\eta(t)$ converge to 0. (For further asymptotic results of linear switched systems with a common non-strict quadratic Lyapunov function, see [6,33].)

The strategy suggested by the previous theorem is the following:

– identify $u_{*,1}, \ldots, u_{*,m}$ such that

  $\cap_{j=1}^m \mathrm{Ker}(A - L(u_{*,j})C) = (0)$;

– impose that each pattern takes all values $u_{*,1}$, …, $u_{*,m}$.

Under these constraints the solution e of (18) is guaranteed to converge to the origin (monotonically with respect to the norm induced by the positive matrix P).

In the case of the metal plate we will see that it is sufficient to take $m = 2$ and that the constraint that each pattern passes trough the two values $u_{*,1}, u_{*,2}$ is not a heavy obstacle in the implementation of the proposed algorithm. As a result, we will obtain a strategy $u(\tilde{x})$ that, under the assumption that the initial state $x(0)$ and the initial estimation $\tilde{x}(0)$ are in R_x and satisfy $\eta(0) < \eta_0$, the trajectory $\mathbf{x}(t, x(0), u)$ and the estimated trajectory, denoted by $\tilde{\mathbf{x}}(t, \tilde{x}(0), u)$, are such that the evaluation of $\mathbf{x}(\cdot)$ after each pattern is again in R_x and $\mathbf{x}(t, x(0), u) - \tilde{\mathbf{x}}(t, x(0), u) \to 0$ as $t \to +\infty$.

7.3 Observer based decomposition

We present here the adaptations of the algorithms taking the observation into account. The *observer based decomposition* algorithm takes η_0 as a new input. Given a system Σ, two sets $R_x \subset \mathbb{R}^n$ and $R_y \subset \mathbb{R}^m$, a positive integer k, and an initial reconstruction error η_0, a successful observer based decomposition returns a set $\tilde{\Delta}$ of the form $\{V_i, Pat_i\}_{i \in I}$, where I is a finite set of indices, every V_i is a subset of R_x, and every Pat_i is a k-pattern such that:

(a) $\bigcup_{i \in I} V_i = R_x$,
(b) for all $i \in I$: $Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$,
(c) for all $i \in I$: $Post_{Pat_i, C}(V_i + \eta_0) \subseteq R_y$.

Such a decomposition allows to perform an output feedback control on Σ as stated in the following. The algorithm relies on two functions given in Algorithms 5 and 6. If a successful observer based decomposition is obtained, it naturally induces an estimate-dependent control, which we denote by $\mathbf{u}_{\tilde{\Delta}}$. By looking for patterns mapping $R_x + \eta_0$ into R_x, we guarantee that $\mathbf{x}(t, x, u)$ is stabilized in R_x. Indeed, if $x(0)$ is the initial state, and $\tilde{x}(0)$ the initial estimation (supposed belonging to R_x), we know that $\tilde{x}(0)$ belongs to V_{i_0} for some $i_0 \in I$, and that $x(0)$ belongs to $V_{i_0} + \eta_0$, so the application of the pattern Pat_{i_0} yields $\mathbf{x}(|Pat_{i_0}|\tau, x(0), Pat_{i_0}) \in R_x - \eta_0$ (because $Post_{Pat_{i_0}}(V_{i_0} + \eta_0) \subseteq R_x - \eta_0$) and $\tilde{\mathbf{x}}(|Pat_{i_0}|\tau, \tilde{x}(0), Pat_{i_0}) \in R_x$ because

$$\|\mathbf{x}(|Pat_{i_0}|\tau, x(0), Pat_{i_0}) - \tilde{\mathbf{x}}(|Pat_{i_0}|\tau, \tilde{x}(0), Pat_{i_0})\| < \eta_0.$$

Note that we plan to improve these algorithms by taking the decrease of $\eta(t)$ into account, so that the decomposition is less restrictive when $\eta(t)$ is small.

7.4 Reduced output feedback control

Algorithms 5 and 6 allow to synthesize guaranteed output feedback controllers for switched control systems without model order reduction. However, the use of model order reduction and observation for the thermal problem of section 6.1 is indeed possible, this is partly enabled thanks to the elliptic nature and highly contractive behavior of the system.

Algorithm 5: Bisection_Obs($W, R_x, R_y, D, K, \eta_0$)

Input: A box W, a box R_x, a box R_y, a degree D of bisection, a length K of input pattern, an initial reconstruction error η_0
Output: $\langle\{(V_i, Pat_i)\}_i, True\rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i + \eta_0) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$, or $\langle \_, False\rangle$

$(Pat, b) := Find\_Pattern(W, R_x, R_y, K, \eta_0)$
if $b = True$ **then**
  return $\langle\{(W, Pat)\}, True\rangle$
else
  if $D = 0$ **then**
    return $\langle \_, False\rangle$
  else
    Divide equally W into $(W_1, \dots, W_{2^n})$
    for $i = 1 \dots 2^n$ **do**
      $(\Delta_i, b_i) :=$ Bisection(W_i,R_x,R_y,$D-1$,K,η_0)
    return $(\bigcup_{i=1\dots 2^n} \Delta_i, \bigwedge_{i=1\dots 2^n} b_i)$

Algorithm 6: Find_Pattern_Obs(W, R_x, R_y, K, η_0)

Input: A box W, a box R_x, a box R_y, a length K of input pattern, an initial reconstruction error η_0
Output: $\langle Pat, True\rangle$ with $Post_{Pat}(W + \eta_0) \subseteq R_x$,$Post_{Pat,C}(W + \eta_0) \subseteq R_y$, or $\langle \_, False\rangle$ when no input pattern maps $W + \eta_0$ into R_x

for $i = 1 \dots K$ **do**
  $\Pi :=$ set of input patterns of length i
  while Π *is non empty* **do**
    Select Pat in Π
    $\Pi := \Pi \setminus \{Pat\}$
    if $Post_{Pat}(W + \eta_0) \subseteq R_x - \eta_0$ *and* $Post_{Pat,c}(W + \eta_0) \subseteq R_y$ **then**
      return $\langle Pat, True\rangle$
return $\langle \_, False\rangle$

The online simulations are performed just as sated in Figure 14. From the full-order system Σ, we build a reduced-order system $\hat{\Sigma}$ by balanced truncation. An ε-decomposition is then performed on $\hat{\Sigma}$, yielding a $\hat{x}$-dependent controller (the decomposition was obtained in about two minutes). The control $u(\tilde{\hat{x}})$ is then computed online with the reconstructed variable $\tilde{\hat{x}}$, which dynamics is the following:

$$\dot{\tilde{\hat{x}}} = \hat{A}\tilde{\hat{x}} - L(u)(\hat{C}\tilde{\hat{x}} - Cx) + \hat{B}u, \quad L(u) \in \mathbb{R}^{n_r \times m} \quad (19)$$

As the ε-decomposition is already quite restrictive (i.e. the error bound overestimates the real projection error) and because the Luenberger observer converges fast, we observe that the induced control already works, even if we do not have any justification of the efficiency yet. The proof should be established by evaluating, for any pattern Pat, a bound of the following error:

$$\|\pi_R \mathbf{x}(|Pat|\tau, x(0), Pat) - \tilde{\hat{\mathbf{x}}}(|Pat|\tau, \tilde{\hat{x}}(0), Pat)\| \quad (20)$$

In the simulations Figures 15 and 16, the full-order system is of order $n = 897$, the reduced order system of order $n_r = 2$. The full-order system is initialized with a uniform temperature field of $x(0) = 0.06^n$. The reduced observer is initialized at $\tilde{x}(0) = 0^2$. The two projected variables $\pi_R x$ cannot be reconstructed exactly because of (at least) the projection error, but the output is still very well reconstructed. Both the observer and the full-order outputs are sent in the objective set R_y, which means that we should manage to control a thermal problem just with the information obtained with few sensors.

8 Final Remarks

Two methods have been proposed to synthesize controllers for switched control systems using model order reduction and the state-space bisection procedure. An offline and an online use are enabled, both uses are efficient but they present different advantages. The offline method allows to obtain the same behavior as the reduced-order model, but the associated bound is more pessimistic, and the controller has to be computed before the use of the real system. The online method leads to less pessimistic bounds but implies a behavior slightly different from the reduced-order model, and the limit cycles may be different from those computed on the reduced system. The behavior of the full-order system is thus less known, but its use can be performed in real time.

A first step to the online reconstruction of the state of the system has been done with the help of Luenberger observers. Numerical simulations seem to show a good behavior with reconstruction and model reduction but the efficiency must still be proved. The use of Kalman filters is however not dismissed.

We are still investigating new model order reductions, more adapted to hyperbolic systems, and with the aim of controlling non linear PDEs. A recent trail which we also want to develop is the dimensionality reduction [19,35,34]. Less restrictive than model order reduction, it should permit to use a fine solver and post-processing techniques to use bisection on a reduced space more representative of the system behavior.

Acknowledgement. This work is supported by Institut Farman through the project SWITCHDESIGN, by the French National Research Agency through the "iCODE Institute project" funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02, and by the Labex DigiCosme ANR-11-LABEX-0045-DIGICOSME.

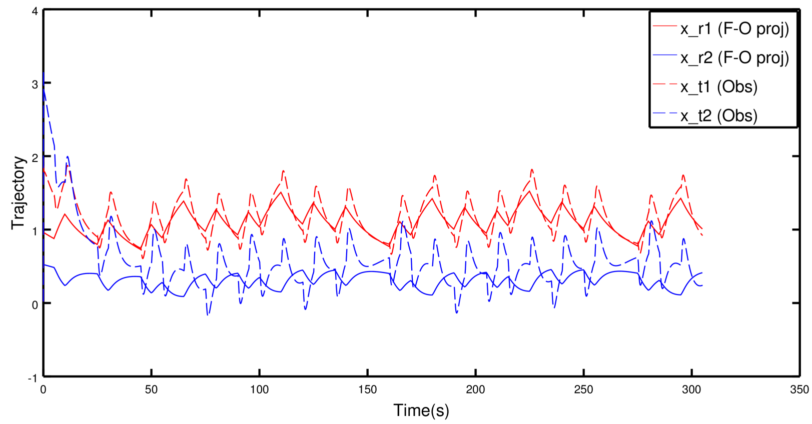

Fig. 15 Simulation of the thermal problem with observation: projected variables. x_r1 and x_r2 are the two variables $\pi_R x$ plotted within time (plain lines), it corresponds to the projection of the full-order system state. x_t1 and x_t2 are the two variables $\tilde{\hat{x}}$ plotted within time (dotted lines), it corresponds to the state of the reduced observer.

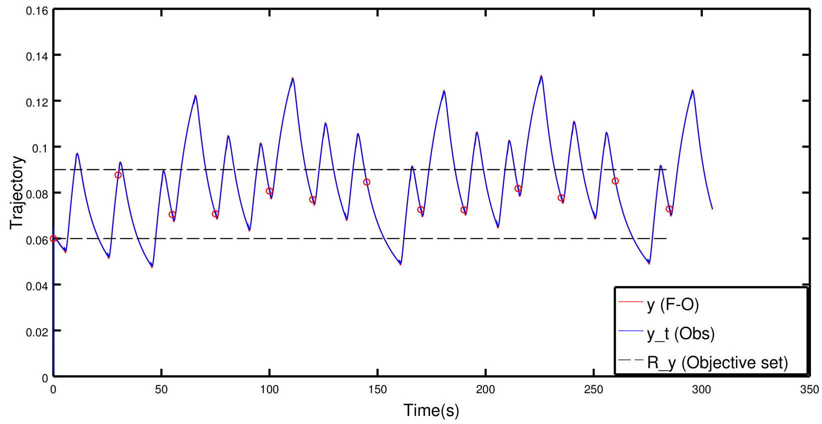

Fig. 16 Simulation of the thermal problem with observation: output variables. The output of the full-order system (plain red) coincides with the output reconstructed by the observer (plain blue), both are sent in the objective set at the end of patterns (red circles).

References

1. A. Alessandri, M. Baglietto, and G. Battistelli. Luenberger observers for switching discrete-time linear systems. *International Journal of Control*, 80(12):1931–1943, 2007.
2. A. Alessandri and P. Coletta. Design of luenberger observers for a class of hybrid linear systems. In *Hybrid systems: computation and control*, pages 7–18. Springer, 2001.
3. A. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: an overview. *International Journal of Applied Mathematics and Computer Science*, 11(5):1093–1121, 2001.
4. A. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2000.
5. M. Azam, S. N. Singh, A. Iyer, and Y.P. Kakad. Nonlinear rotational maneuver and vibration damping of nasa scole system. *Acta Astronautica*, 32(3):211–220, 1994.
6. M. Balde and P. Jouan. Geometry of the limit sets of linear switched systems. *SIAM J. Control Optim.*, 49(3):1048–1063, 2011.
7. P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale lyapunov equations, riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.
8. P. Benner and A. Schneider. Balanced truncation model order reduction for lti systems with many inputs or outputs. In *Proceedings of the 19th international symposium on mathematical theory of networks and systems–MTNS*, volume 5, 2010.
9. F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.
10. F. Chinesta, P. Ladeveze, and E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, 2011.
11. W. Clark. Vibration control with state-switched piezoelectric materials. *Journal of intelligent material systems and structures*, 11(4):263–271, 2000.
12. L. Cordier and M. Bergmann. Proper Orthogonal Decomposition: an overview. In *Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics, 2008.*, page 46 pages. VKI, 2008.
13. F. Formosa. Contrôle actif des vibrations des structures type poutre. *Private report*.

14. L. Fribourg, U. Kühne, and R. Soulat. Finite controlled invariants for sampled switched systems. *Formal Methods in System Design*, 45(3):303–329, December 2014.
15. L. Fribourg and R. Soulat. *Control of Switching Systems by Invariance Analysis: Application to Power Electronics*. Wiley-ISTE, July 2013. 144 pages.
16. L. Fribourg and R. Soulat. Stability controllers for sampled switched systems. In Parosh Aziz Abdulla and Igor Potapov, editors, *Proceedings of the 7th Workshop on Reachability Problems in Computational Models (RP'13)*, volume 8169 of *Lecture Notes in Computer Science*, pages 135–145, Uppsala, Sweden, September 2013. Springer.
17. A. Girard. Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 111–120. ACM, 2010.
18. A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *Automatic Control, IEEE Transactions on*, 55(1):116–126, Jan 2010.
19. R. Gunawan, E. L. Russell, and R. D. Braatz. Comparison of theoretical and computational characteristics of dimensionality reduction methods for large-scale uncertain systems. *Journal of Process Control*, 11(5):543–552, 2001.
20. N. W. Hagood and A. von Flotow. Damping of structural vibrations with piezoelectric materials and passive electrical networks. *Journal of Sound Vibration*, 146:243–268, April 1991.
21. Z. Han and B. H. Krogh. Reachability analysis of hybrid systems using reduced-order models. In *American Control Conference*, pages 1183–1189. IEEE, 2004.
22. Z. Han and B. H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 287–301. Springer Berlin Heidelberg, 2006.
23. M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013.
24. H. Ji, J. Qiu, H. Nie, and L. Cheng. Semi-active vibration control of an aircraft panel using synchronized switch damping method. *International Journal of Applied Electromagnetics and Mechanics*, 46(4), 2014.
25. G. Kerschen, J.-C. Golinval, A. Vakakis, and L. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear Dynamics*, 41(1-3):147–169, 2005.
26. A. Le Coënt, F. de Vusyt, C. Rey, L. Chamoin, and L. Fribourg. Guaranteed control of switched control systems using model order reduction and state-space bisection. *Open Acces Series in Informatics*, 2015.
27. M. Mazo Jr., A. Davitian, and P. Tabuada. Pessoa: A tool for embedded controller synthesis. In T. Touili, B. Cook, and P. Jackson, editors, *Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 566–569. Springer Berlin Heidelberg, 2010.
28. S.O. R. Moheimani and A.J. Fleming. *Piezoelectric transducers for vibration control and damping*. Springer Science & Business Media, 2006.
29. B. C. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE Transaction on Automatic Control*, 26(1), 1981.
30. S. Mouelhi, A. Girard, and G. Goessler. CoSyMA: a tool for controller synthesis using multi-scale abstractions. In *HSCC'13 - 16th International Conference on Hybrid systems: computation and control*, pages 83–88, Philadelphie, United States, April 2013. ACM.
31. R. Nong and D. C. Sorensen. A parameter free adi-like method for the numerical solution of large scale lyapunov equations. 2009.
32. A. Ramaratnam, N. Jalili, and D. M. Dawson. Semi-active vibration control using piezoelectric-based switched stiffness. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5461–5466. IEEE, 2004.
33. P. Riedinger, M. Sigalotti, and J. Daafouz. On the algebraic characterization of invariant sets of switched linear systems. *Automatica J. IFAC*, 46(6):1047–1052, 2010.
34. S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
35. A. A. H. Salah, T. Garna, J. Ragot, and H. Messaoud. Synthesis of a robust controller with reduced dimension by the loop shaping design procedure and decomposition based on laguerre functions. *Transactions of the Institute of Measurement and Control*, page 0142331215583101, 2015.
36. U. Serres, J.-C. Vivalda, and P. Riedinger. On the convergence of linear switched systems. *IEEE Transactions on Automatic Control*, 56(2):320–332, 2011.
37. S. M. Shahruz. Boundary control of the axially moving kirchhoff string. *Automatica*, 34(10):1273–1277, 1998.
38. P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.
39. K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, pages 2323–2330, 2002.
40. M. Zeitz. The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2):149–156, 1987.