C. R. Mecanique 347 (2019) 793–805

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Data-Based Engineering Science and Technology / *Sciences et technologies de l'ingénierie basées sur les données*

Nonintrusive data-based learning of a switched control heating system using POD, DMD and ANN

Tarik Fahlaoui, Florian De Vuyst *

Laboratoire de mathématiques appliquées de Compiègne EA 2222, Université de technologie de Compiègne, Alliance Sorbonne Université, 60200 Compiègne, France

ARTICLE INFO

Article history:
Received 11 July 2019
Accepted 31 October 2019
Available online 14 November 2019

Keywords:
Data-driven model
Heating system
Switched control
Heat equation
Model order reduction
POD
DMD
ANN
Machine learning

ABSTRACT

The aim of this work is to derive an accurate model of two-dimensional switched control heating system from data generated by a Finite Element solver. The nonintrusive approach should be able to capture both temperature fields, dynamics and the underlying switching control rule. To achieve this goal, the algorithm proposed in this paper will make use of three main ingredients: proper orthogonal decomposition (POD), dynamic mode decomposition (DMD) and artificial neural networks (ANN). Some numerical results will be presented and compared to the high-fidelity numerical solutions to demonstrate the capability of the method to reproduce the dynamics.

1. Introduction

Switched control systems are commonly used in engineering applications such as in the automotive ([1], [2]) or network management. The discrete control is built from a small set of control modes indexed by some integers, and the state of the system evolves according to the heat equation with a particular heat source for each control mode. The system is closed by a switching control rule that selects the right mode during time evolution. Generally, the rule is defined in order to provide stability of the global system with states lying into an invariant domain. The flexibility in the definition of the switching rule makes switched control systems interesting for many applications and in different situations.

Theoretical questions of stability or reachability have been deeply studied in the literature, especially for switched linear systems in both discrete form (when differential equations are discretized) or continuous form ([3], [4]). An important aspect of these switched linear systems is that they show a global nonlinear behavior (with a local linear behavior for each fixed mode).

The accurate learning of nonlinear dynamical systems is still nowadays a difficult task, and there is no definitive way to achieve it. Recent works about *Dynamic Mode Decomposition* (DMD) and *Extended DMD* (EDMD, [5]) added with Koopman theory to approximate nonlinear dynamical systems by transforming them into (quasi)linear dynamical systems are a promising research direction. However, the methodology requires the definition of an a priori dictionary of functions such that the space spanned by these functions correctly approximates the Koopman eigenfunctions, and so far there is unfortu-

* Corresponding author.
E-mail addresses: tarik.fahlaoui@utc.fr (T. Fahlaoui), florian.de-vuyst@utc.fr (F. De Vuyst).

https://doi.org/10.1016/j.crme.2019.11.005

nately no general rule to construct it in an optimal way. Some authors have proposed the use of artificial neural networks (ANN, [6]) or recurrent neural network (RNN, [7]), but generally this requires big data in the training process, which is not suitable in our context because of the prohibitive computational time of simulation data. Specific issues of ANN like the phenomenon of *vanishing gradient* and *exploding gradient* possibly make the network training phase low or even nonconvergent ([8]). Another difficulty arises since we are looking for spatial PDE-based problems, and thus once the spatial discretization have been done, the state of the semi-discretized system of ODEs lives in a high-dimensional vector space (the size of the state is the number of degrees of freedoms, which can be of order one million or more). Some dimensionality reduction strategies (POD, EIM, PGD etc.) have to be included into the global methodology.

1.1. Related work

The machine learning of switched systems has been recently addressed in the context of robotics ([9]), and follows some works where the action of the robot onto its environment is learned for better performance purposes. In Lee et al. [9], the dynamics of the system is nonlinear but the state space is low-dimensional. Learning phase was done by means of a Gaussian process. The mode transition is learned by the use of piecewise smooth transition functions. For the present work, the temperature field is globally smooth, so it seems hard to directly detect the phase transition since jumps or sharp changes do not really occur in the solution. Still in [9] the authors use numerous trajectories to classify the dynamic modes and to learn them. In contrast, in our case it is preferable to learn the dynamics with only one spatiotemporal full-order simulation (still because of heavy computational costs). From the full order model we have access to the state of controller during time (heating is activated or not), but we do not know the heating rule and we want to retrieve the correlation between the controller mode and the temperature field. In PDE's context, Peitz et al. ([10]) accelerate the search of an optimal controller, by first transforming the control problem into a switched control problem, and then by using the Koopman operator theory of dynamical system and pointwise evaluations of the solution in order to derive a reduce order model for each autonomous system. In Proctor et al. [11], the learning of a control system is managed by separating the action of the dynamic system onto the state and the action of the control. By this way it is easier to understand how the control acts. This work is done in the context of real data, moreover it requires to perform two singular value decompositions (SVD) on the measured states. In the present work we use simulation data that include Finite Element temperature fields at different sample times and the history of heating states (array of boolean values). For the temperature fields, we will also make use of a suitable reduced basis (classical POD-based approach) for dimensionality reduction.

In the next section, the problem of heating control is introduced and formulated. It will be then implemented into a FE code used as a data generator. The switching rule of interest is a hysteresis controller (also known as bang-bang controller). This kind of controller is commonly used in mechanical and electrical engineering.

1.2. Organization of the paper

As mentioned above the main goal of this article is to achieve a nonintrusive learning of this switching control heating system from data generated by a full-order FE solver. For the sake of clarity and for reproducibility purposes, we have decided to detail all the steps of the process, i.e. from the FE solver and data generation to the learning process itself. Section 1 will present the switched control heat problem and the Finite Element scheme in use for data generation. In Section 2 we give some clarification on the simulation results used as data in the learning process. Section 4 is devoted the identification of the dynamical system for each control mode; we will make use of DMD for that. The section 5 is dedicated to the machine learning of the switching rule. Because it is highly nonlinear, it seems suitable to use ANNs. The ANN structure used for this application will be justified. We will finally end up with the section 6 dedicated to the numerical experiments. Some numerical results will be presented and compared to the high-fidelity numerical solutions to demonstrate the capability of the method to reproduce the complex dynamics. Concluding remarks and perspectives are given into Section 7.

2. Problem setting and high-fidelity solver

Let us consider the case of a two-dimensional medium heated by an electrical resistance. A sensor S is used to measure the temperature on a particular point of the plate, denoted by $\mathbf{x}_S$. The objective is to control the temperature in such a way that it stays between the bounds $T_{\min}$ and $T_{\max}$. The heat loss through the domain boundary is expressed by a heat flux ϕ_{out}. Both temperatures $T_{\min}$ and $T_{\max}$ are calibrated in such a way that they can be attained in the presence or absence of heating. This control is composed of two modes, one for an activated mode (heating of the electrical resistance) and one for the disable mode. The switching rule will depend on sensor temperature $T(\mathbf{x}_S)$. We have a boolean heating state denoted by $\mathcal{H}(t)$ at current time t.

Schematic 1 briefly summarizes the above switching problem. Mathematically, the problem can be formulated as a coupled system made of a partial differential heat problem and a heating state rule governed by an algebraic equation. The heat problem can be written

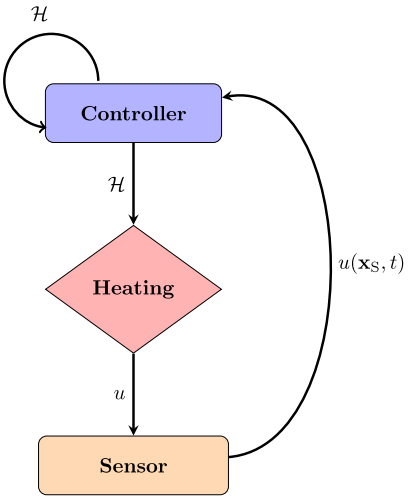


Fig. 1. Schematic of the switching control heat problem.

$$\begin{cases} \partial_t u - \nabla \cdot (\kappa(.)\nabla u) = f & \text{in } \Omega \times (0,T) \\ \kappa\, \partial_{\mathbf{n}} u = -\phi_{\text{out}} & \text{on } \partial\Omega \times (0,T) \\ u(\mathbf{x},0) = T_{\min} & \text{in } \Omega \end{cases} \tag{1}$$

where Ω denotes the plate, κ is the thermal conductivity in the plate domain, equal to κ_R in the resistance zone $\Omega_R \subset \Omega$ and κ_P otherwise, and f is the control term. Usually, in the switched control community, the notation $f_{\sigma(t)}$ stands for the control term, where $\sigma(t)$ denoted the selected control index at time t. However in our application there is only one control term since if the plate is not activated, the source term is switched to zero, so it is simpler to denote it by $f = f(\cdot, t)$ only. The source term f is designed as follows:

$$f(\mathbf{x}, t) = q(\mathbf{x})\mathcal{H}(t) \tag{2}$$

with the heat source function

$$q(\mathbf{x}) = q_R\, \chi_{\Omega_R}(\mathbf{x}) \tag{3}$$

for a given constant $q_R > 0$, the initial heating state $\mathcal{H}(0) = \mathcal{H}_0 \in \{0, 1\}$ and the transition rule

$$\mathcal{H}(t) \leftarrow 0 \quad \text{if } (u(\mathbf{x}_S, t) = T_{\max}) \text{ and } (\mathcal{H}(t) = 1) \tag{4}$$

$$\mathcal{H}(t) \leftarrow 1 \quad \text{if } (u(\mathbf{x}_S, t) = T_{\min}) \text{ and } (\mathcal{H}(t) = 0) \tag{5}$$

The function $q(\mathbf{x})$ is the heat flux emitted by the electrical resistance which is constant over Ω_R and is zero otherwise.

It is known that solutions of heat problems have good regularity properties even for weakly regular boundary conditions or source terms (at least continuity by Sobolev embedding in $\mathbb{R}^2$). So we can expect that the switching rule activates transitions only at sparse times.

Remark 1. The switched control system's study belongs to the class of the *repetitive control* (RC), where the desired action onto the system is the same over time. The learning of an optimal control for repetitive control has been investigated intensively, especially in the robotics context (iterative learning control ILC, see [12], [13], [14]).

2.1. High fidelity finite element solver

In this small section we present the *high-fidelity* Finite Element solver used to discretize the heat equation presented in the latter section. The nature of the heat equation (1) is parabolic. For that reason a backward Euler scheme is used in order to achieve unconditional stability with a time step δt that does not depend on the spatial mesh size. The semi-discrete problem in time thus reads: given a discrete solution $u^n \in H^1(\Omega)$ at time $t^n = n\delta t$, find $u^{n+1} \in H^1(\Omega)$ at time $t^{n+1} = (n+1)\delta t$ solution to the problem

$$\begin{cases} \dfrac{u^{n+1}}{\delta t} - \nabla \cdot \left(\kappa(.)\nabla u^{n+1}\right) = \dfrac{u^n}{\delta t} + q_R\, \chi_{\Omega_R} \mathcal{H}(u^n) & \text{in } \Omega \\ \kappa\, \partial_{\mathbf{n}} u^{n+1} = -\phi_{\text{out}} & \text{on } \partial\Omega \end{cases} \tag{6}$$

At initial time, one can use for example a uniformly constant temperature field $u^0 = T_{\min}$ and assume that the system is in heating mode ($\mathcal{H}_0 = 1$). Hence, we have to solve a Poisson-type equation for each time step. The finite element method is suitable for this kind of problem, and allow us to deal with different geometries, moreover the problem (6) is classical and require no special finite element spaces. Therefore we choose the classical P^1 Lagrange finite elements with FE hat basis functions $\hat{\phi}_i(\mathbf{x})$ to achieve spatial discretization, and we denote by u_h^n the spatial approximation of u^n. A discrete variational formulation of the problem (6) is set in the FE discrete space. We will denote by $\mathbf{u}^n$ the vector containing all the degrees of freedom (in our case it is nothing else but the nodal values of $u_h^n(.)$), and by N_x its size. Starting from $\mathbf{u}^0$, the successive fields $\mathbf{u}^n$ will be computed for each $n \in \{1, ..., N_t\}$ with $N_t = \frac{T}{\Delta t}$. At discrete time t^n, for a known vector $\mathbf{u}^n$, we search for a vector $\mathbf{u}^{n+1}$ at time t^{n+1} solution of the system of equations

$$\mathcal{M}\mathbf{u}^{n+1} + \delta t \mathcal{K}\mathbf{u}^{n+1} = \delta t \mathcal{H}(\mathbf{u}^n)\mathcal{Q} - \delta t \Phi_{\text{out}} + \mathcal{M}\mathbf{u}^n$$

i.e.

$$\mathbf{u}^{n+1} = (\mathcal{M} + \delta t\, \mathcal{K})^{-1}\mathcal{M}\mathbf{u}^n + \delta t\, \mathcal{H}(\mathbf{u}^n)(\mathcal{M} + \delta t\, \mathcal{K})^{-1}\mathcal{Q} - \delta t\, (\mathcal{M} + \delta t\, \mathcal{K})^{-1}\Phi_{\text{out}} \tag{7}$$

where $\mathcal{M}$ is the FE mass matrix, $\mathcal{K}$ is the regular FE stiffness matrix, the vector $\mathcal{Q}$ has components $\mathcal{Q}_i = q_R \int_{\Omega_R} \hat{\phi}_i(\mathbf{x})\, \mathrm{d}\mathbf{x}$, and the discrete boundary heat flux vector Φ_{out} is such that $\left(\Phi_{\text{out}}\right)_i = \int_{\partial\Omega} \phi_{\text{out}} \hat{\phi}_i(\mathbf{x})\, \mathrm{d}\mathbf{x}$. The second term in the RHS in (7) depends on the heating state $\mathcal{H}(\mathbf{u}^n)$, which makes equation (7) nonlinear. However, since the heating state is constant during a time step δt, the mapping that transforms $\mathbf{u}^n$ into $\mathbf{u}^{n+1}$ is linear. Actually, since there are only two discrete states in this system ($\mathcal{H}(t) \in \{0, 1\}$), the whole dynamics can be described by two independent linear dynamics plus a nonlinear controller that switches from a linear dynamics to the other. The equation (7) can be written in a more condensed form

$$\mathbf{u}^{n+1} = \mathcal{A}\mathbf{u}^n + \mathcal{B}^\nu \tag{8}$$

where $\mathcal{A} = (\mathcal{M} + \delta t\, \mathcal{K})^{-1}\mathcal{M}$ and $\mathcal{B}^\nu$ is the source term that includes heats source and boundary heat flux terms, with dependency on the heating states $\mathcal{H}(\mathbf{u}^n)$ represented by a boolean variable $\nu \in \{0, 1\}$.

To summarize, the *Full Order Model* (FOM) (8) is used to generate some data (details will be given in the next section). The following section will be devoted to the construction of a *Reduced Order Model* (ROM) from this data.

3. Data generation

Once the time-dependent problem (1) has been solved by the full-order solver, we save the spatio-temporal data and store them in a matrix $\mathbf{U}$. The columns of $\mathbf{U}$ are the different vectors $\mathbf{u}^n$, $n = 0, ..., N_t$. Each vector $\mathbf{u}^n$ has a large size, thus it will be necessary to construct a reduced basis (RB) for dimensionality reduction, then project the snapshots and compute the reduced order projection coefficients. In order to achieve it, we will apply a Principal Component Analysis (PCA) on these data: first we need to compute the correlation matrix S of elements

$$\mathcal{S}_{n,m} = \int_\Omega u_h^n(x)\, u_h^m(x)\, \mathrm{d}x$$

by the use of full-order model and save this matrix. Next, to learn the switching rule, the heating state of the system has to be known during time. Thus, we collect the states $\mathcal{H}^n = \mathcal{H}(t^n)$ at each discrete time t^n and store them in a column vector $\mathbf{H}$.

The two matrices $\mathbf{U}$, S and the vector $\mathbf{H}$ will compose the dataset used for the learning of the ROM.

4. Construction of a low dimensional space for model order reduction

The discrete vector $\mathbf{u}^n$ lives in the high-dimensional finite element space. The need of a reduced basis is motivated by the following observations:

1) the identification of the mapping that transforms each $\mathbf{u}^n$ into $\mathbf{u}^{n+1}$ would be hard to determine because of too many unknowns and because of a limited dataset;
2) usually, N_x is big compared to the number of time steps N_t, this makes the learning problem ill-posed as we will see later.
3) a low-dimensional model of (7) is preferable for computational performance reasons.

The standard way to achieve this is to use an offline procedure in order to find a low-dimensional vector space W^K of dimension K from a snapshot of the solutions $\mathbf{u}^n$ computed by the finite-element solver during the time. For example, we can search to minimize the distance between each vectors $\mathbf{u}^n$ and its projection $\Pi^K\mathbf{u}^n$ over W^K, i.e. by solving the least-squares problem

$$\min_{(K,W^K)} \quad \frac{1}{2}\sum_{n=0}^{N_t} \|\mathbf{u}^n - \Pi^K \mathbf{u}^n\|^2$$

subject to the accuracy criterion

$$\frac{\sum_{n=0}^{N_t} \|\mathbf{u}^n - \Pi^K \mathbf{u}^n\|^2}{\sum_{n=0}^{N_t} \|\mathbf{u}^n\|^2} \leq \varepsilon$$

for a certain tolerance $\varepsilon > 0$, $\varepsilon \ll 1$. In other word, we will look for an orthonormal set $\left(\boldsymbol{\phi}_k\right)_{1\leq k\leq K}$ of $\mathbb{R}^{N_x}$, with $K \ll N_t$, such that, for each $\mathbf{u}^n$, there exists a linear combination of $\boldsymbol{\phi}_k$ that approximates the vector $\mathbf{u}^n$ in an accurate way. The vector made of the projection coefficients will then be the low-order representation of the initial vector.

A well-known common method is the *Proper Orthogonal Decomposition* (POD, [15]), also known as *Principal Component Analysis* (PCA, [16]) in Statistics or *Karhunen–Loève* (KL, [17]) in probability theory. The idea is to consider a set of snapshots (here the discrete solutions $\mathbf{u}^n$ at each time step t^n), and perform an eigenvalue decomposition onto the correlation matrix S, whose elements are

$$\mathcal{S}_{n,m} = \int_\Omega u_h^n(\mathbf{x}) u_h^m(\mathbf{x})\,\mathrm{d}\mathbf{x} = \left(\mathbf{u}^n\right)^\intercal \mathcal{M}\mathbf{u}^m = \langle \mathbf{u}^n, \mathbf{u}^m\rangle_{\mathcal{M}}$$

Consider the diagonal decomposition of $\mathcal{S}$, $\mathcal{S} = \mathbf{V}\Lambda\mathbf{V}^\intercal$, where Λ is a diagonal matrix of some real eigenvalues λ_k and $\mathbf{V}$ the orthogonal matrix of eigenvectors $\mathbf{v}_k \in \mathbb{R}^{N_t}$. Assuming that the eigenvalues λ_k of $\mathcal{S}$ are arranged in decreasing order, we keep only the K first dominating eigenvalues with their eigenvectors by setting

$$\mathbf{V}_r = \mathbf{V}_{\cdot,1\cdots K} \quad \text{and} \quad \Lambda_r = \Lambda_{1\cdots K,1\cdots K}$$

The matrix $\boldsymbol{\Phi}_r$ formed by the orthormal set $\left(\boldsymbol{\phi}_k\right)_{1\leq k\leq K}$ is then obtained by the formula

$$\boldsymbol{\Phi}_r = \mathbf{U}\mathbf{V}_r\Lambda_r^{-\frac{1}{2}}$$

Finally, noting that

$$\boldsymbol{\Phi}_r^\intercal \mathcal{M}\mathbf{U} = \Lambda_r^{-\frac{1}{2}}\mathbf{V}_r^\intercal\mathbf{U}^\intercal\mathcal{M}\mathbf{U} = \Lambda_r^{-\frac{1}{2}}\mathbf{V}_r^\intercal\mathcal{S} = \Sigma_r\mathbf{V}_r^\intercal$$

where $\Sigma_r = \Lambda_r^{\frac{1}{2}}$, we found the low-dimensional representations of $\mathbf{U}$

$$\left[\,\Xi^1 \cdots \Xi^{N_t}\,\right] = \Sigma_r\mathbf{V}_r^\intercal$$

In the literature, this coefficients vectors are usually called the POD coefficients.

The projection of each vector $\mathbf{u}^n$ onto W^K reads

$$\Pi^K\mathbf{u}^n = \sum_{k=1}^{K}\left(\Xi^n\right)_k\boldsymbol{\phi}_k \tag{9}$$

The Eckart–Young–Mirsky theorem gives us the following estimate

$$\frac{\sum_{n=0}^{N_t}\|\mathbf{u}^n - \Pi^K\mathbf{u}^n\|^2}{\sum_{n=0}^{N_t}\|\mathbf{u}^n\|^2} = \frac{\left(\sum_{k>K}\lambda_k\right)^{\frac{1}{2}}}{\left(\sum_{k=1}^{N_t}\lambda_k\right)^{\frac{1}{2}}}$$

and the optimality criterion

$$\frac{1}{2}\sum_{n=0}^{N_t}\|\mathbf{u}^n - \Pi^K\mathbf{u}^n\|^2 \leq \min_{\mathrm{rank}(\mathbf{V})=K}\frac{1}{2}\sum_{n=0}^{N_t}\|\mathbf{u}^n - \mathbf{v}^n\|^2$$

The more redundant information will be contained in $\mathbf{U}$, the smaller will be K. In our case the spatial operator $-\nabla\cdot(\kappa\nabla\cdot)$ has a regularizing effect on the solution, thus we could expect that the information in the matrix $\mathbf{U}$ is poor. However the switching process during time evolution adds some information entropy.

5. Dynamics identification procedure

This section is devoted to the identification of the dynamics. We would like to find a simple (linear) model of the form

$$\mathbf{u}^{n+1} = \mathcal{A}\,\mathbf{u}^n \qquad \forall n \in \{1, \cdots, N_t\}$$

The *Dynamic Mode Decomposition* (DMD) method was originally proposed for such kind of problems using Krylov spaces ([18]), and a new definition of the DMD method was given using the Penrose–Moore pseudo-inverse ([19]). Below we will present a slightly different approach of the DMD method that will be more suitable to our problem.

5.1. The standard DMD method

As an introduction, let us search the matrix $\mathcal{A}$ as a solution to a least-squares problem. We define the two training sets

$$\mathcal{X} = [\mathbf{u}^1 \cdots \mathbf{u}^{N_t-1}] \quad \text{and} \quad \mathcal{Y} = [\mathbf{u}^2 \cdots \mathbf{u}^{N_t}]$$

where $\mathcal{X}$ is the input training dataset and $\mathcal{Y}$ the output dataset. Since the equivalence

$$\mathbf{u}^{n+1} = \mathcal{A}\,\mathbf{u}^n, \quad \forall n \in \{1, \cdots, N_t - 1\} \iff \mathcal{A}\,\mathcal{X} = \mathcal{Y}$$

holds, we look for the matrix $\mathcal{A}^*$ defined by

$$\mathcal{A}^* = \arg\min_{\mathcal{A}} J(\mathcal{A}) := \frac{1}{2}\sum_{n=1}^{N_t} \|\left(\mathcal{A}\mathcal{X} - \mathcal{Y}\right)^n\|^2$$

The functional J is quadratic (w.r.t. $\mathcal{A}$), hence the optimality conditions implicitly define the matrix $\mathcal{A}^*$ as the solution to the following matrix equation

$$\mathcal{A}^*\,\mathcal{X}\mathcal{X}^\intercal = \mathcal{Y}\mathcal{X}^\intercal$$

If the matrix $\mathcal{X}\mathcal{X}^\intercal$ were invertible, we would have an explicit expression for $\mathcal{A}^*$. Actually it is not the case, because $\mathcal{X}$ is a $N_x \times N_t$ rectangular matrix, and thus the rank of $\mathcal{X}\mathcal{X}^\intercal$ is less or equal to N_t. In fact, there is non-uniqueness of the solution to this minimization problem. Consequently, we temporarily add a Tykhonov regularization term to the functional J, and we define the matrix $\mathcal{A}^*_\alpha$ by

$$\mathcal{A}^*_\alpha = \arg\min_{\mathcal{A}} J_\alpha(\mathcal{A}) := \frac{1}{2}\sum_{n=1}^{N_t} \|\left(\mathcal{A}\mathcal{X} - \mathcal{Y}\right)^n\|^2 + \frac{\alpha}{2}\|\mathcal{A}\|_F^2$$

This minimization problem has a unique solution, which is given by

$$\mathcal{A}^*_\alpha = \mathcal{Y}\mathcal{X}^\intercal\left(\mathcal{X}\mathcal{X}^\intercal + \alpha\mathbf{I}\right)^{-1}$$

Passing to the limit when $\alpha \to 0$, we recover the Moore–Penrose pseudo-inverse definition, which is

$$\mathcal{X}^\dagger := \lim_{\alpha\to 0} \mathcal{X}^\intercal\left(\mathcal{X}\mathcal{X}^\intercal + \alpha\mathbf{I}\right)^{-1} = \lim_{\alpha\to 0}\left(\mathcal{X}^\intercal\mathcal{X} + \alpha\mathbf{I}\right)^{-1}\mathcal{X}^\intercal$$

And finally, since the matrix $\mathcal{X}^\intercal\mathcal{X}$ is invertible we get

$$\mathcal{A}^* = \mathcal{Y}\mathcal{X}^\dagger$$

Adding a Tykhonov regularization term to the functional J settled the non-uniqueness of the learning problem. However, it gives us the matrix with the lowest 2-norm spectrum.

Another convenient way to settle non-uniqueness is to work in a low-dimensional POD space, then to identify the dynamics from the POD coefficients. Roughly speaking, we will not look for a mapping that transforms $\mathbf{u}^n$ into $\mathbf{u}^{n+1}$, but for a mapping that transforms Ξ^n into Ξ^{n+1}. Hence, we consider the new training datasets

$$X = [\Xi^1 \cdots \Xi^{N_t-1}] \quad \text{and} \quad Y = [\Xi^2 \cdots \Xi^{N_t}]$$

As soon as $K < N_t - 1$, the matrix $XX^\intercal$ is invertible, and we get the matrix

$$A = Y X^\dagger$$

and we refer to this method as the *standard DMD method*.

5.2. A more suitable variant DMD approach

The DMD approach could be used to identify the dynamics of the switched control system. However, we can derive a more accurate and suitable DMD model according to the following remarks. As shown in section 1 (equation (8)), the FOM is of the form

$$\mathbf{u}^{n+1} = \mathcal{A}\,\mathbf{u}^n + \mathcal{B}^\nu \tag{10}$$

with a source term $\mathcal{B}^\nu$ that depends on the heating state $\nu = \nu(\mathbf{u}^n)$. As explained in the previous section, the standard DMD method is a linear model of the form

$$\mathbf{a}_r^{n+1} = \mathcal{A}_r\,\mathbf{a}_r^n \tag{11}$$

Hence the surrogate model (11) cannot handle the RHS in an optimal way, and has a poor description of the switching feature as well as the non-linearity. In order to derive a more accurate ROM, we proceed as follows:

- the RHS is captured by increasing the dimension of the input training set X (in the same spirit that the one used in Extended DMD, EDMD);
- the switch dependency into the RHS is handled by clustering the training datasets according to the values of the vector $\mathbf{H}$ ($\nu = 0$ and $\nu = 1$);
- the non-linearity hidden in the index ν is approximated by using an Artificial Neural Network (ANN).

Thus, we define the two sets of integers:

$$\mathcal{N}_1 := \{n \text{ such that } \mathbf{H}_n = 1\} \quad \text{and} \quad \mathcal{N}_0 := \{m \text{ such that } \mathbf{H}_m = 0\}$$

We then build two training data sets

$$X_1 = \begin{bmatrix} \Xi^{n_1} \cdots \Xi^{n_{\#(\mathcal{N}_1)}} \\ 1 \cdots 1 \end{bmatrix}, \qquad Y_1 = [\Xi^{n_1+1} \cdots \Xi^{n_{\#(\mathcal{N}_1)}+1}]$$

$$X_0 = \begin{bmatrix} \Xi^{m_1} \cdots \Xi^{m_{\#(\mathcal{N}_0)}} \\ 1 \cdots 1 \end{bmatrix}, \qquad Y_0 = [\Xi^{m_1+1} \cdots \Xi^{m_{\#(\mathcal{N}_0)}+1}]$$

Finally, we get the two augmented matrices

$$\begin{pmatrix} A^\nu & B^\nu \end{pmatrix} = Y_\nu X_\nu^\dagger \quad \text{with } \nu = \{0, 1\}$$

The ROM will be then in the form

$$\mathbf{a}_r^{n+1} = \begin{pmatrix} A^\nu & B^\nu \end{pmatrix} \begin{bmatrix} \mathbf{a}_r^n \\ 1 \end{bmatrix} = A^\nu\,\mathbf{a}_r^n + B^\nu \tag{12}$$

for each controlled mode ν.

Remark 2. Even if we expect that $A^0 = A^1$ (same low-order heating operator), it is not imposed in this method.

The ROM (12) allows us to quickly predict the behavior of the temperature on the plate during time evolution. However we still need to identify the switching control rule. This issue will be treated in the next section.

5.3. The DMD method as a neural network

The DMD method presented in section 4 can also be viewed as a feedforward artificial neural network (ANN) with a single layer and no rectifier (a linear activation function) for the output. Indeed, the weights correspond to the coefficients of the matrices A^ν and the biases to the vectors B^ν. The Fig. 2 shows the neural network architecture corresponding to the DMD method.

6. Learning of the switched control

Neural networks and deep learning method currently know important developments because of their powerfulness and capability to learn nonlinear complex systems from data. This section is devoted to the learning of the nonlinear switching control rule. Let us first show that the switching rule in use can be reproduced by a simple 3-layer ANN. The control selects the states of the electrical resistance according to the temperature sensor and the previous state. Since the output is a boolean which corresponds to a switching mode (0 or 1), this kind of problem could be viewed as a supervised

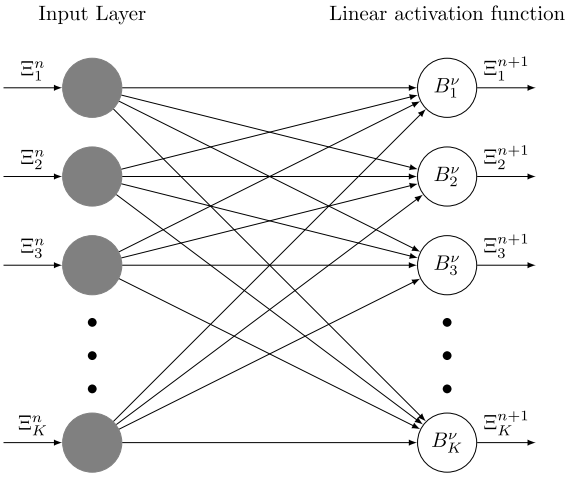


Fig. 2. The DMD model as a one-layer ANN.

classification problem ([20]) where the predictor features are the previous state and the temperature sensor, and the label the current state of the electrical resistance.

Several classifiers are commonly used, among them we can cite the *Decision Tree* ([21]), the *Support-Vector Machines* (SVM, [22]), the *k-Nearest Neighbors* (kNN, [23]), or the *Artificial Neural Networks* (ANN, [24]).

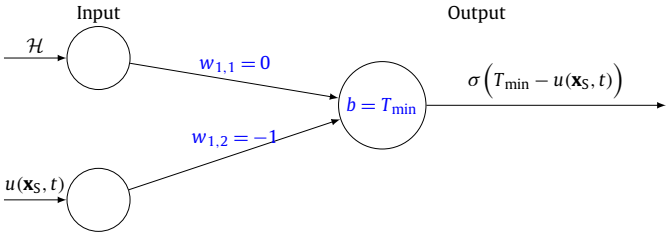
However, since the controller here performs some simple tests to select the switching mode, it seems more natural to use ANNs. It is supported by the proposition below.

Proposition 1. *The switching control rule $\mathcal{H}$ could be written as a feedforward neural network with a 3-2-1 structure.*

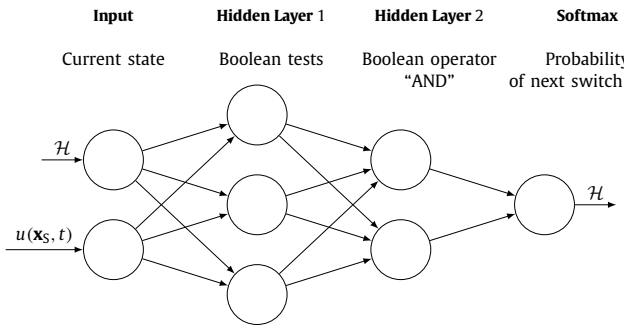
Proof. We first recall the definition of the function $\mathcal{H}$

$$\mathcal{H}(t) \leftarrow 0 \quad \text{if } (u(\mathbf{x}_S, t) > T_{\max}) \text{ and } (\mathcal{H}(t) = 1)$$
$$\mathcal{H}(t) \leftarrow 1 \quad \text{if } (u(\mathbf{x}_S, t) < T_{\min}) \text{ and } (\mathcal{H}(t) = 0)$$

The test $u(\mathbf{x}_S, t) < T_{\min}$ could be done by the function below



Extending this basic test, the switching control rule $\mathcal{H}$ could be written with the following feedforward neural network



where the operations at each neuron are as follows.

Input. The input will be the temperature at the sensor ($u(\mathbf{x}_S, t)$) and the previous switching rule ($\mathcal{H}$).
First hidden layer. Three tests are done:
- $u(\mathbf{x}_S, t) < T_{\min}$
- $u(\mathbf{x}_S, t) > T_{\max}$.
- Is the electrical resistance activated?

Second hidden layer. Intersection of the hyperplans (boolean operation "AND").
Output. The probability of switch state at the next discrete time. □

Remark 3. This result ensures that we could write the switching control rule as a "minimal" feedforward neural network. Hence, it justifies the use of an ANN, as long as we use sigmoid activation functions for the hidden layers, and a softmax function for the output layer. Some comparisons with other Machine Learning classifiers for the switching control rule learning are presented in [25].

For the computation, we use a feedforward neural network with only one hidden layer, and with 5 neurons in the hidden layer. The output activation function will be still a `softmax` function. It could be noted that this architecture does not require any knowledge about the switching control rule.

7. Numerical results

We now present some numerical results in order to demonstrate both accuracy and stability of the approach, for both short-range and long-range time behavior of the model. We will first detail the computational aspects of data generation. Then, we will show the identifying each subsystem dynamics, and the learning of the switching rule. Some numerical comments will be discussed, especially for the neural network. Finally, a numerical spectral analysis of the ROM structure will be achieved.

7.1. Data generation

All this work deals with synthetic data generated by the high-fidelity FE solver. The physical and numerical parameters used, and the geometry of the problem are depicted in Fig. 3. The numerical computations will be done thanks to the flexible software FreeFem++ ([26]). Once the FE solver has computed the temperature and the correlation matrix $\mathcal{S}$, the data will be sent to the ROM software platform (namely Matlab in our experiments).

The numerical solutions from the FE solver are depicted in Fig. 4 (a) for different times. The ON/OFF switching control makes this problem highly dynamic. Moreover, it could be noted that the temperature has a periodic behavior, this is due to the repetitive aspect of the control. The evolution of the sensor temperature $u(\mathbf{x}_S, \cdot)$ is depicted in Fig. 4 (b) with corresponding state history. To quantify the accuracy of the ROM prediction, we use the 400 first time steps for the training and the 400 last time steps will be used for the validation.

7.2. Training and prediction from the ROM

In this paragraph, we will quantify the quality of the ROM first for the training process and then for the prediction. First, we measure the accuracy of the ROM from the training data set using the following error estimate:

$$\mathcal{E}_{L^2}^{\text{train}} := \sqrt{\sum_{n=0}^{N_{t,\text{train}}} \|\Phi_r \mathbf{a}_r^n - \mathbf{u}^n\|_{\mathcal{M}}^2} \Big/ \sqrt{\sum_{n=0}^{N_{t,\text{train}}} \|\mathbf{u}^n\|_{\mathcal{M}}^2}$$

Second, the prediction capability of the ROM is evaluated from the validation data set using the second error estimate:

$$\mathcal{E}_{L^2}^{\text{valid}} := \sqrt{\sum_{n=N_{t,\text{train}}+1}^{N_{t,\text{valid}}} \|\Phi_r \mathbf{a}_r^n - \mathbf{u}^n\|_{\mathcal{M}}^2} \Big/ \sqrt{\sum_{n=N_{t,\text{train}}+1}^{N_{t,\text{valid}}} \|\mathbf{u}^n\|_{\mathcal{M}}^2}$$

where $\mathbf{a}_r^n$ are the POD coefficients provided by the ROM, which is

$$\mathbf{a}_r^{n+1} = \begin{pmatrix} A^\nu & B^\nu \end{pmatrix} \begin{bmatrix} \mathbf{a}_r^n \\ 1 \end{bmatrix} = A^\nu \mathbf{a}_r^n + B^\nu$$

$$\tilde{\mathbf{u}}^{n+1} = \Phi_r \mathbf{a}_r^n$$

$$\tilde{\mathbf{u}}^{n+1}(\mathbf{x}_S) = \sum_{k=1}^{K} \boldsymbol{\phi}_k(\mathbf{x}_S) \left(\mathbf{a}_r^{n+1}\right)_k, \quad \nu \leftarrow \nu(\tilde{\mathbf{u}}^{n+1}(\mathbf{x}_s), \nu^n)$$

Parameter	Value
Thermal conductivity (κ)	0.2 W·m^{-1}·K^{-1}
Minimal temperature ($T_{\min}$)	20 °C
Maximal temperature ($T_{\max}$)	21 °C
Electrical resistance's flux (q)	4 W·m^{-2}
External flux (Φ_{out})	0.5 W·m^{-1}
Number of iterations for the training ($N_{t,\text{train}}$)	400
Number of iterations for the validation ($N_{t,\text{valid}}$)	800
Time step (δt)	4
Number of mesh vertices (N_x)	4997

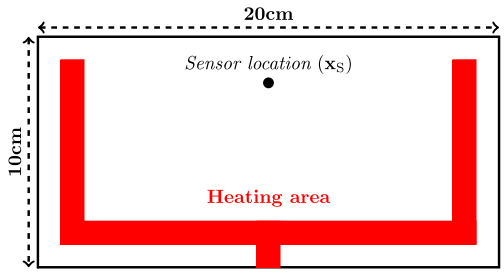
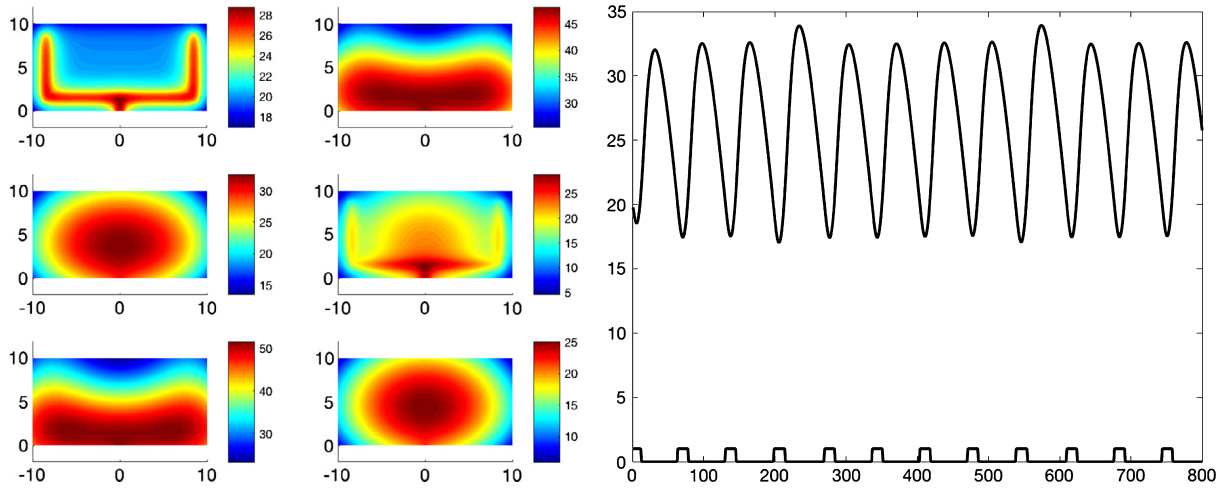


Fig. 3. Geometry and parameters used for data generation.



(a) Temperature on the plate at different times.

(b) Sensor temperature and electrical resistance states during time evolution.

Fig. 4. High-fidelity FEM solver outputs.

Table 1
Training and prediction errors for the temperature provided by the ROM.

#POD modes	$\mathcal{E}_{L^2}^{\text{train}}$			$\mathcal{E}_{L^2}^{\text{valid}}$	Relative POD error	$\mathcal{E}_{\text{DMD}}$
	DMD+rule	DMD+ANN	DMD+rule	DMD+ANN		
$K=1$	0.3422	0.3426	0.3447	0.3442	0.1431	6.7989E−04
$K=5$	0.0885	0.0644	0.0719	0.0719	0.0031	1.6289E−04
$K=10$	2.5696E−04	2.5696E−04	2.1409E−04	2.1409E−04	2.8840E−04	5.0326E−04
$K=20$	2.1581E−04	2.1581E−04	1.7340E−04	1.7341E−04	2.5607E−04	0.0011
$K=30$	2.0798E−04	2.0797E−04	1.6542E−04	1.6541E−04	2.4951E−04	0.0014
$K=60$	2.1021E−04	2.1021E−04	1.3723E−04	1.3723E−04	2.2921E−04	0.0018

The matrices A^ν and B^ν are computed by the DMD method (10), whereas the nonlinear function ν is learned by the use of the ANN discussed above. Thus we will consider the case where we use the analytic expression of the control (cf $\mathcal{H}$ expression) and the case where we use the ANN in order to analyze independently the performance of these two methods. In the table below, will be also depicted the estimator $\mathcal{E}_{\text{DMD}}$ defined by

$$\mathcal{E}_{\text{DMD}} := \frac{1}{\|Y\|_F}\left\{\sum_{\nu=1}^{2}\|\left(A^\nu \;\; B^\nu\right)X_\nu - Y_\nu\|_F^2\right\}^{\frac{1}{2}}$$

which is the least-squares error, and quantify the correlation learned from the DMD approach. Finally, we will then take different values of K.

The results shown in Table 1 confirms the good accuracy of the ROM for a reasonable number of POD modes ($K > 10$), we even obtain the full-order model accuracy. Moreover, the ROM has a same accuracy onto the validation data set, which is not obvious since the learning was done without this data and furthermore, the POD coefficients are computed by an iterative process, thus if a small error is committed, the temperature obtained by the ROM will explode ad infinitum. When the number of POD coefficients is small, the relative POD error is important, which causes a poor quality of the learning

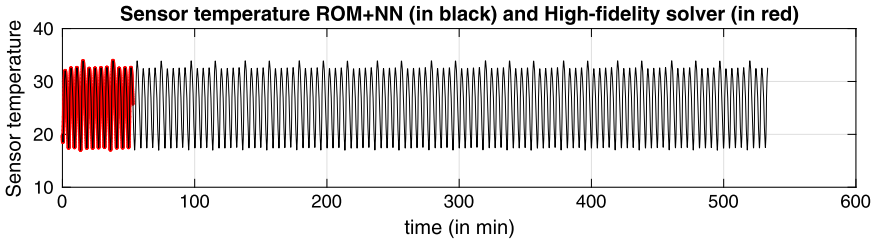


Fig. 5. Long-time behavior of the sensor temperature provided by the ROM+ANN (in black) and high-fidelity solver sensor temperature (in red).

Table 2
Computational time.

Model	CPU time (s)
FOM	57.077
ROM+rule	0.060
ROM+ANN	0.119

Table 3
ROM accuracy according to the rate of learning data used.

Rate of learning data used (learning data set size)	$\mathcal{E}_{L^2}^{\text{train}}$		$\mathcal{E}_{L^2}^{\text{valid}}$	
	DMD+rule	DMD+ANN	DMD+rule	DMD+ANN
6.25% (50)	0.0613	0.0054	0.3056	0.2954
12.5% (100)	3.2058E−04	3.2058E−04	2.3849E−04	0.1174
25% (200)	2.7401E−04	0.0104	1.8157E−04	0.2201
37.5% (300)	2.3770E−04	2.3770E−04	1.7791E−04	1.7790E−04
50% (400)	2.1581E−04	2.1581E−04	1.7340E−04	1.7341E−04
75% (600)	2.0724E−04	2.0724E−04	1.6871E−04	1.6871E−04

data. However, from the DMD formula, we get two matrices that minimize the least-squares error between the input and the output training set. And the fact that the estimator $\mathcal{E}_{\text{DMD}}$ is small means that we learn non-physical correlations.

Since the dimension of the training set is small, we train our neural network using a quasi-Newton method, moreover the initialization of the weights and the biases has an important influence on the optimization process. In order to avoid some local minima and to increase the score of the neural network on unknown data set, we do cross-validation. Switching control rule learning will be done in Python by using the library *Scikit-learn* ([27]). As depicted in Table 1, the ROM+ANN gives temperature with the same accuracy as the one obtained with the knowledge of the rule. We have thus derived a high accurate non-intrusive ROM. Moreover the ROM seems to be asymptotically stable, as it is shown in Fig. 5 where we plot the sensor temperature given by the ROM+ANN.

A last attention could be given to the computational time since the main goal of this work is to derive an inexpensive surrogate model. The table below presents the computational time between the FOM, the ROM with the rule and the ROM with the ANN. So we get a speed-up of order 500 between the FOM and the DMD+ANN ROM. (See Table 2.)

7.3. Influence of the training data set size

In all the previous computations, we have used 50% of the data for the training. However we can wonder what happen if we use less data. To answer this question, we consider different numbers of time steps for the learning process, and then give the results in the table below. All the computations were done with $K = 20$.

The more data available for the learning stage the more accurate will be our ROM, thus it is natural that the error decreases with the increasing rate of data used. Furthermore, Table 3 indicates that the need of data is more important for the neural network training than for DMD.

7.4. Spectral properties of the ROM

We finally close this section with a spectral analysis of the ROM. As soon as we have computed a linear mapping from the input training data set to the output training data set, we could perform an eigendecomposition of this mapping for spectral property analysis. Checking the spectrum property can also inform us about a correct learning of the Physics or not. The model (1) is the heat equation which is inherently dissipative. This implies that the temperature profile will converge to zero without external excitation or non-homogeneous boundary condition. This property is fundamental, and we expect that our ROM can capture it. The figure below show the spectrum of the two mapping A^0 (which corresponds to the OFF state) and A^1 (for the ON state) in the complex plane. From the ROM form (12), we could remark that if the eigenvalues are

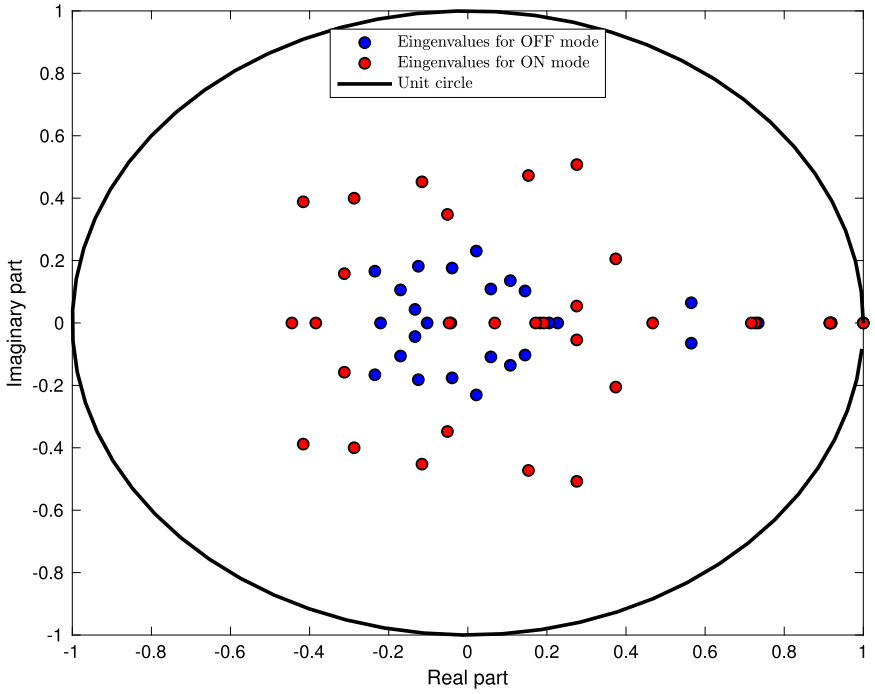


Fig. 6. Eigenvalues of the matrices A^0 (in blue), for the electrical resistance OFF mode, and A^1 (in red), for the ON mode, on the unit circle (in black).

in the unit circle it mean that certain quantities are conserved during time evolution, if at least one eigenvalue is outside of the unit disk the temperature will explode, and finally if all the eigenvalues are inside of the unit disk the temperature goes to zero. In order to be coherent with the model (1), we expect that all the eigenvalues are inside of the unit disk. (See Fig. 6.)

This is the case, we have thus correctly captured the physical nature of the problem. It could be noted that the imaginary part is not zero for all eigenvalues which denotes a lack of symmetry.

8. Concluding remarks

The identification of a switching control system from data has been made possible by a combination of POD, DMD and ANN.

We have been able to achieve an inexpensive, asymptotically stable, and high accurate reduced order model from simulation data. Furthermore, the use of DMD-type method allows for an easy interpretation of the ROM with available spectral properties. This method could be applied to a broad range of switching control systems.

References

[1] M.A. Ahmad, S.-I. Azuma, I. Baba, T. Sugie, Switching controller design for hybrid electric vehicles, SICE J. Control, Meas., Syst. Integr. 7 (5) (2014) 273–282.
[2] A. Ferrara, J. Paderno, Application of switching control for automatic pre-crash collision avoidance in cars, Nonlinear Dyn. 46 (3) (2006) 307–321, https://doi.org/10.1007/s11071-006-9044-x.
[3] D. Liberzon, Switching in Systems and Control, Springer Science & Business Media, 2003.
[4] H. Lin, P.J. Antsaklis, Stability and stabilizability of switched linear systems: a survey of recent results, IEEE Trans. Autom. Control 54 (2) (2009) 308–322.
[5] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data–driven approximation of the Koopman operator: extending dynamic mode decomposition, J. Nonlinear Sci. 25 (6) (2015) 1307–1346.
[6] Q. Li, F. Dietrich, E.M. Bollt, I.G. Kevrekidis, Extended dynamic mode decomposition with dictionary learning: a data-driven adaptive spectral decomposition of the Koopman operator, Chaos, Interdiscip. J. Nonlinear Sci. 27 (10) (2017) 103111.
[7] B.A. Pearlmutter, Learning state space trajectories in recurrent neural networks, Neural Comput. 1 (2) (1989) 263–269.
[8] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013, pp. 1310–1318.
[9] G. Lee, Z. Marinho, A.M. Johnson, G.J. Gordon, S.S. Srinivasa, M.T. Mason, Unsupervised learning for nonlinear piecewise smooth hybrid systems, arXiv preprint, arXiv:1710.00440, 2017.
[10] S. Peitz, S. Klus, Koopman operator-based model reduction for switched-system control of pdes, arXiv preprint, arXiv:1710.06759, 2017.
[11] J.L. Proctor, S.L. Brunton, J.N. Kutz, Dynamic mode decomposition with control, SIAM J. Appl. Dyn. Syst. 15 (1) (2016) 142–161.
[12] D. Tharayil, A.G. Alleyne, A survey of iterative learning control: a learning-based method for highperformance tracking control, IEEE Control Syst. Mag. 26 (3) (2006) 96–114.

[13] D.H. Owens, K. Feng, Parameter optimization in iterative learning control, Int. J. Control 76 (11) (2003) 1059–1069.
[14] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, J. Robot. Syst. 1 (2) (1984) 123–140.
[15] S. Volkwein, Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling, Lecture Notes, vol. 4, University of Konstanz, 2013.
[16] I. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.
[17] A. Alexanderian, A brief note on the Karhunen-Loève expansion, arXiv preprint, arXiv:1509.07526, 2015.
[18] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.
[19] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S.L. Brunton, J.N. Kutz, On dynamic mode decomposition: theory and applications, arXiv preprint, arXiv: 1312.0041, 2013.
[20] S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: a review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. 160 (2007) 3–24.
[21] S.K. Murthy, Automatic construction of decision trees from data: a multi-disciplinary survey, Data Min. Knowl. Discov. 2 (4) (1998) 345–389, https://doi.org/10.1023/A:1009744630224.
[22] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (2) (1998) 121–167.
[23] P. Cunningham, S.J. Delany, k-nearest neighbour classifiers, Mult. Classif. Syst. 34 (8) (2007) 1–17.
[24] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533.
[25] T. Fahlaoui, Réduction de modèles et apprentissage de solutions spatio-temporelles paramétrées, PhD thesis, Université de Technologie de Compiègne, France, 2020, Submitted (defense in Jan 2020).
[26] F. Hecht, New development in freefem++, J. Numer. Math. 20 (3–4) (2012) 251–265.
[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.