49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Reduction of the computational burden of POD models with polynomial nonlinearities

Oscar Mauricio Agudelo, Jairo José Espinosa, *Senior Member, IEEE,*
and Bart De Moor, *Fellow Member, IEEE*

***Abstract*— This paper presents a technique for making the evaluation of POD models with polynomial nonlinearities less intensive. Regularly, Proper Orthogonal Decomposition (POD) and Galerkin projection have been employed to reduce the high-dimensionality of the discretized systems used to approximate Partial Differential Equations (PDEs). Although a large model-order reduction can be obtained with these techniques, the computational saving during simulation is small when we have to deal with nonlinear or Linear Time Variant (LTV) models. In this paper, we present a method that exploits the polynomial nature of POD models derived from input-affine high-dimensional systems with polynomial nonlinearities, for generating compact and efficient representations that can be evaluated much faster. Furthermore, we show how the use of the feature selection techniques can lead to a significant computational saving.**

I. INTRODUCTION

Proper Orthogonal Decomposition (POD), also known as Karhunen-Loève expansion or Principal Component Analysis (PCA), is a technique that has been applied in many physical systems modeled by Partial Differential Equations (PDEs) for deriving reduced order models. POD is a data driven method where a suitable set of orthonormal basis vectors are derived from simulation or experimental data. These basis vectors, which are organized in order of relevance, capture the spatial dynamics of the systems. The reduced order models are typically obtained by projecting (Galerkin projection) the high-dimensional models on the most relevant basis vectors.

Although we can get a large model-order reduction with the POD technique, the computational saving offered by the reduced order models is small when nonlinear or Linear Time Variant (LTV) systems are considered. The reason of this limitation lies in the fact that we need the full spatial information from the original high-dimensional systems in order to evaluate the reduced order models. In [1]–[3] a method known as Missing Point Estimation (MPE) is proposed for tackling this problem. In this method the Galerkin projection is conducted only on some pre-selected state variables instead of the entire set. The remaining state variables are estimated by means of the POD basis vectors. It has been reported that this technique can save considerable computational effort. In [4], a Multi-Layer Perceptron (MLP) neural network is used to approximate the nonlinear vector function of the POD model of a heat transfer process in a one-dimensional bar. Given that the time for evaluating the trained MLP can be very short, we can save a substantial amount of time. However one of the main drawbacks of this approach is the local minima problem during the training of the neural network.

In this paper we propose a method for speeding up the evaluation of nonlinear POD models derived from input-affine high-dimensional systems with polynomial nonlinearities. It turns out that by exploiting the polynomial nature of these POD models, we can construct compact and efficient representations that require less computational power. Besides, by using sequential feature selection algorithms for choosing their most relevant monomials (suboptimal solution), we can noticeably increase the computational saving.

This paper is organized as follows. Section II presents a description of the dynamical system that will be used to explain our technique, the nonlinear heat transfer problem in a one-dimensional bar that was used in [4]. In Section III the derivation of a reduced order model for the bar by means of POD and Galerkin projection is discussed. Section IV introduces our technique for accelerating the nonlinear POD model found in Section III by exploiting its polynomial nature. Section V shows some validation and simulation results, and finally in Section VI we present some concluding remarks.

II. HEAT TRANSFER IN A ONE-DIMENSIONAL BAR

The system under study is the silicon bar shown in Figure 1. The bar has attached an actuator which provides a uniformly distributed heat flux $u(t)$ between $z = z_{\mathrm{a}}$ and $z = z_{\mathrm{b}}$. Additionally, an external heat flux $d(t)$ is applied uniformly along the bar whose ends are kept at $25^{\circ}\mathrm{C}$ (ambient temperature) all the time.

If only temperature variations in the z-direction are considered, the dynamics of the temperature $T(z,t)$ of the bar can be modeled by the following nonlinear PDE:

$$\rho C_{\mathrm{p}} \frac{\partial T(z,t)}{\partial t} = \frac{\partial}{\partial z}\left(\kappa\left(T(z,t)\right)\frac{\partial T(z,t)}{\partial z}\right) + V(z,t) \quad (1)$$

with the following initial and boundary conditions: $T(z, t = 0) = 25^{\circ}\mathrm{C}$, $T(z = 0, t) = T(z = L, t) = 25^{\circ}\mathrm{C}$. Here ρ is the material density in $[\mathrm{Kg} \cdot \mathrm{m}^{-3}]$, C_{p} is the heat capacity

O. M. Agudelo is with the Department of Electrical Engineering (ESAT), Research Group SCD-SISTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium. He is also with the department of Automation and Electronics, Universidad Autónoma de Occidente, Calle 25 No 115-85, Cali, Colombia. E-mail: mauricio.agudelo@esat.kuleuven.be

J. J. Espinosa is with the Facultad de Minas, Universidad Nacional de Colombia, Carrera 80 No. 65-223, Medellín, Colombia. E-mail: jairo.espinosa@ieee.org

B. De Moor is with the Department of Electrical Engineering (ESAT), Research Group SCD-SISTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium. E-mail: bart.demoor@esat.kuleuven.be

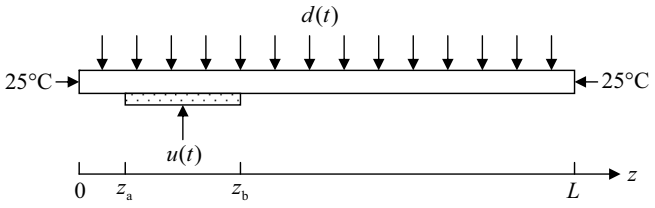

Fig. 1. Silicon Bar.

in $[\mathrm{J}\cdot\mathrm{Kg}^{-1}\cdot\mathrm{K}^{-1}]$, $\kappa(T)$ is the temperature dependent heat conductivity in $[\mathrm{J}\cdot\mathrm{s}^{-1}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}]$, t is the time in [s], z is the spatial coordinate in [m] and $V(z,t)$ is the heat source applied to the bar at position z and time t in $[\mathrm{W}\cdot\mathrm{m}^{-3}]$. $V(z,t)$ is defined as follows:

$$V(z,t) = \begin{cases} d(t)+u(t), & z_\mathrm{a} \le z \le z_\mathrm{b} \\ d(t), & \text{elsewhere.} \end{cases}$$

The relation between the temperature and the heat conductivity $\kappa(T)$ is approximated by a polynomial of degree 3,

$$\kappa(T) = \kappa_0 + \kappa_1 T + \kappa_2 T^2 + \kappa_3 T^3 \tag{2}$$

where $\kappa_0 = 36$, $\kappa_1 = -0.1116$, $\kappa_2 = 1.7298\times10^{-4}$ and $\kappa_3 = -1.78746\times10^{-7}$ are real coefficients in the appropriated units. The length of the bar is $L = 0.1$ m and the remaining numerical values of the model parameters are: $\rho = 3970\ \mathrm{Kg}\cdot\mathrm{m}^{-3}$, $C_\mathrm{p} = 766\ \mathrm{J}\cdot\mathrm{Kg}^{-1}\cdot\mathrm{K}^{-1}$, $z_\mathrm{a} = 0.005$ m and $z_\mathrm{b} = 0.04$ m. The operating ranges in $[\mathrm{W}\cdot\mathrm{m}^{-3}]$ of $d(t)$ and $u(t)$ are $-500\cdot10^3 \le d(t) \le 500\cdot10^3$ and $-1500\cdot10^3 \le u(t) \le 1500\cdot10^3$ respectively. Some of the previous numerical values were inspired on the values given in [5].

For simulation purposes, it is necessary to reduce the infinite dimensionality of (1) by discretizing the spatial domain. To this end, the partial derivatives with respect to space were replaced by backward (the inner spatial derivative) and forward (the outer spatial derivative) difference approximations. The discretized model of the bar is given by the following set of ordinary differential equations:

$$\frac{dT_i}{dt} = c_1\Big(\kappa(T_{i+1})T_{i+1} - \big(\kappa(T_{i+1})+\kappa(T_i)\big)T_i + \kappa(T_i)T_{i-1}\Big) + c_2 V_i \tag{3}$$

$$c_1 = \frac{1}{\rho C_\mathrm{p}(\Delta z)^2},\quad c_2 = \frac{1}{\rho C_\mathrm{p}},\quad T_0 = T_N = 25^\circ\mathrm{C}$$

$$\text{for } i = 1,\ldots,N-1,$$

where N is the number of sections in which the bar is divided, Δz is the length of each section, and T_i and V_i are the temperature and heat flux at the point $z_i = i\Delta z$.

If we define $\mathbf{T}(t) \in \mathbb{R}^{N-1} = [T_1(t),\ldots,T_{N-1}(t)]^T$ as the vector containing the temperature of the grid points from z_1 till z_{N-1} every time instant, we can write equation (3) as follows:

$$\dot{\mathbf{T}}(t) = \mathbf{F}(\mathbf{T}(t)) + \mathbf{B}_1 d(t) + \mathbf{B}_2 u(t) \tag{4}$$

where $\mathbf{F}(\mathbf{T}(t)) : \mathbb{R}^{N-1} \to \mathbb{R}^{N-1}$ is a vector-valued or vector function which contains the nonlinear terms of the model, and $\mathbf{B}_1$ and $\mathbf{B}_2$ are vectors defined as: $\mathbf{B}_1 \in \mathbb{R}^{N-1} = [c_2, c_2, \ldots, c_2]^T$, $\mathbf{B}_2 \in \mathbb{R}^{N-1} = [0,\ldots,0,c_2,\ldots,c_2,0,\ldots,0]^T$. The position of the nonzero elements in $\mathbf{B}_2$ corresponds to the position of the grid points that are in contact with the actuator.

The spatial domain was divided into $N = 500$ sections which means that (4) has $N-1 = 499$ states. With such amount of states the design of a control system for the bar is not an easy task. In addition, the simulation of (4) demands a considerable amount of computational resources. In order to tackle this situation, in the next section a reduced order model (few number of equations and states) of the bar will be obtained by means of POD and Galerkin projection.

III. Nonlinear POD model of the system

Given that the initial state of (4) does not provide information about the system dynamics, we are going to work with the temperature deviations with respect to the ambient temperature (25°C). Consequently, the vector $\mathbf{T}(t)$ is split as follows:

$$\mathbf{T}(t) = \mathbf{T}^\Delta(t) + \mathbf{T}^*$$

where $\mathbf{T}^\Delta(t) \in \mathbb{R}^{N-1}$ is the vector containing the deviations of the temperature profile and $\mathbf{T}^* \in \mathbb{R}^{N-1}$ is a constant vector which contains the initial temperature profile of the bar (ambient temperature).

In POD, we start by observing that $\mathbf{T}^\Delta(t) \in \mathbb{R}^{N-1}$ can be expanded as a sum of orthonormal basis vectors:

$$\mathbf{T}^\Delta(t) = \sum_{j=1}^{N-1} a_j(t)\boldsymbol{\varphi}_j \tag{5}$$

where $\boldsymbol{\varphi}_j \in \mathbb{R}^{N-1}\ \forall j = 1,\ldots,N-1$ is a set of orthonormal basis vectors (POD basis vectors or POD basis functions) in the discretized spatial domain, and $a_j(t) \in \mathbb{R}\ \forall j = 1,\ldots,N-1$ are the time-varying coefficients, or POD coefficients, associated to each basis vector. These POD basis vectors are ordered according to their relevance to $\mathbf{T}^\Delta(t)$.

The main dynamics of the system can be represented using the first n most relevant basis vectors, since $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_n$ condensates the main spatial correlations. An nth order approximation of (5) is then given by means of the truncated sequence

$$\mathbf{T}^\Delta_n(t) = \sum_{j=1}^{n} a_j(t)\boldsymbol{\varphi}_j,\quad n \ll N-1. \tag{6}$$

An approximate (reduced order) model of $\mathbf{T}^\Delta(t)$ can be derived by building a model for the first n POD coefficients. This is the essence of model reduction by POD.

The POD basis vectors are determined from simulation or experimental data of the process. The dynamic model for the first n POD coefficients is commonly found by means of the Galerkin projection [1]. For the specific case of the linear systems, subspace identification techniques can be used for

finding such a model [6]. In order to make this paper self-contained, in the next subsections we show the derivation of the POD model of the bar as it was presented in [4].

A. Generation of the snapshot Matrix

We have built the snapshot matrix $\mathbf{T}_{\text{snap}} \in \mathbb{R}^{499\times 2001}$ by collecting the evolution of the deviations of the temperature profile when Pseudo Random Multilevel Noise Signals (PRMNS) were applied to the process inputs $u(t)$ and $d(t)$,

$$\mathbf{T}_{\text{snap}} = \left[\mathbf{T}^{\Delta}(0), \mathbf{T}^{\Delta}(\Delta t), \ldots, \mathbf{T}^{\Delta}\left((N_{\text{d}}-1)\Delta t\right)\right].$$

Here, $N_{\text{d}} = 2001$ is the number of samples gathered using a sampling time Δt of 1 s. A commutation probability of 3% was set for the signals and the amplitudes in $[\text{W}\cdot\text{m}^{-3}]$ of $d(t)$ and $u(t)$ were restricted to the intervals $[-500\cdot 10^3 \;\; 500\cdot 10^3]$ and $[-1500\cdot 10^3 \;\; 1500\cdot 10^3]$ respectively.

B. Derivation of the POD basis Vectors

The POD basis vectors were derived by calculating the singular value decomposition (SVD) of $\mathbf{T}_{\text{snap}}$,

$$\mathbf{T}_{\text{snap}} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$$

where $\mathbf{\Phi} \in \mathbb{R}^{499\times 499}$ and $\mathbf{\Psi} \in \mathbb{R}^{2001\times 2001}$ are unitary matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{499\times 2001}$ is a matrix containing the singular values of $\mathbf{T}_{\text{snap}}$ in a decreasing order on its main diagonal. The left singular vectors, that is the columns of $\mathbf{\Phi}$,

$$\mathbf{\Phi} \in \mathbb{R}^{499\times 499} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{499}]$$

are the POD basis vectors.

C. Selection of the most relevant POD basis vectors

We have made the selection by checking the singular values of $\mathbf{T}_{\text{snap}}$, the larger the singular value the more relevant the basis vector is. In this problem, the first $n = 6$ basis vectors were selected. The 6th order approximation of $\mathbf{T}^{\Delta}(t)$ is given by

$$\mathbf{T}_n^{\Delta}(t) = \sum_{j=1}^{n} a_j(t)\boldsymbol{\varphi}_j = \mathbf{\Phi}_n\mathbf{a}(t), \tag{7}$$

where $\mathbf{\Phi}_n = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_6]$ and $\mathbf{a}(t) = [a_1(t), \ldots, a_6(t)]^T$.

D. Construction of the model for the first $n = 6$ POD coefficients

In order to derive a dynamic model for the POD coefficients, we have used the Galerkin projection. If we define a residual function $R(\mathbf{T}, \dot{\mathbf{T}})$ for equation (4) as follows:

$$R(\mathbf{T}, \dot{\mathbf{T}}) = \dot{\mathbf{T}}(t) - \mathbf{F}(\mathbf{T}(t)) - \mathbf{B}_1 d(t) - \mathbf{B}_2 u(t) \tag{8}$$

and we replace $\mathbf{T}(t)$ by its nth order approximation $\mathbf{T}_n(t) = \mathbf{T}_n^{\Delta} + \mathbf{T}^*$ in (8), the projection of $R(\mathbf{T}_n, \dot{\mathbf{T}}_n)$ on the space spanned by the basis vectors $\mathbf{\Phi}_n$ shall vanish. That is,

$$\left\langle R(\mathbf{T}_n, \dot{\mathbf{T}}_n), \boldsymbol{\varphi}_j \right\rangle = 0; \;\; j = 1, \ldots, n = 6 \tag{9}$$

where $\langle .,. \rangle$ denotes the Euclidean inner product. If we replace $\mathbf{T}(t)$ by its nth order approximation $\mathbf{T}_n(t) = \mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*$ in equation (4), and we apply the inner product criterion (9) to the resulting equation,

$$\mathbf{\Phi}_n^T\mathbf{\Phi}_n\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T\mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right) + \mathbf{\Phi}_n^T\mathbf{B}_1 d(t) + \mathbf{\Phi}_n^T\mathbf{B}_2 u(t)$$

then we can find the reduced order model of the bar with only 6 states,

$$\dot{\mathbf{a}}(t) = \mathbf{\Phi}_n^T\mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{10}$$

where $\tilde{\mathbf{B}}_1 = \mathbf{\Phi}_n^T\mathbf{B}_1$ and $\tilde{\mathbf{B}}_2 = \mathbf{\Phi}_n^T\mathbf{B}_2$. Finally if we define a new vector-valued function $\mathbf{f} : \mathbb{R}^6 \to \mathbb{R}^6$ as $\mathbf{f}\left(\mathbf{a}(t)\right) = \mathbf{\Phi}_n^T\mathbf{F}\left(\mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*\right)$, then the reduced order model of the bar can be written more compactly as follows:

$$\dot{\mathbf{a}}(t) = \mathbf{f}\left(\mathbf{a}(t)\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \tag{11a}$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n\mathbf{a}(t) + \mathbf{T}^*. \tag{11b}$$

In general, it should be clear that we do not know the compact expression of $\mathbf{f}(\mathbf{a}(t))$ in (11a). So, in order to simulate the reduced order model, the ODE solver has to evaluate indirectly $\mathbf{f}(\mathbf{a}(t))$. Firstly, the solver has to map the state of the reduced order model $\mathbf{a}(t)$ into the original high-dimensional space by means of this linear transformation $\mathbf{T}_n^{\Delta}(t) = \mathbf{\Phi}_n\mathbf{a}(t)$. Secondly it has to evaluate the resulting high-dimensional state vector $\mathbf{T}_n^{\Delta}(t)$ in the vector function $\mathbf{F}(\mathbf{T}_n^{\Delta}(t) + \mathbf{T}^*)$ of (4), and finally it has to map the results of this evaluation to the low dimensional space by pre-multiplying them by $\mathbf{\Phi}_n^T$. The evaluation of $\mathbf{f}(\mathbf{a}(t))$ is done as many times as it is required by the ODE solver within each integration step. Hence the indirect evaluation of $\mathbf{f}(\mathbf{a}(t))$ is the bottleneck that limits severely the computational gain of the nonlinear POD model. In the following section we propose an alternative approach to the ones presented in [1]–[4], in order to speed up the evaluation of the POD model (11) by exploiting its polynomial nature.

IV. Polynomial POD model

We start by noticing that the mth component function of $\mathbf{f}(\mathbf{a}(t)) = \mathbf{\Phi}_n^T\mathbf{F}(\mathbf{T}_n(t))$ can be written as

$$f_m(\mathbf{a}(t)) = \sum_{i=1}^{N-1} \varphi_{i,m} F_i(\mathbf{T}_n(t)) \tag{12}$$

where $\varphi_{i,m}$ is the ith entry of the basis vector $\boldsymbol{\varphi}_m$, $N = 500$ is the number of sections in which the bar was divided, $n = 6$ is the number of selected basis vectors, $\mathbf{T}_n(t)$ is the nth order approximation of $\mathbf{T}(t)$ and F_i is the ith component function of $\mathbf{F}(\mathbf{T}_n(t))$ defined as follows:

$$F_i(\mathbf{T}_n(t)) = c_1\Big(\kappa(\tilde{T}_{i+1}(t))\tilde{T}_{i+1}(t) - \big(\kappa(\tilde{T}_{i+1}(t)) + \kappa(\tilde{T}_i(t))\big)\tilde{T}_i(t) + \kappa(\tilde{T}_i(t))\tilde{T}_{i-1}(t)\Big), \tag{13}$$

where c_1 is a constant whose definition is given in (3), $\kappa(T) : \mathbb{R} \to \mathbb{R}$ is the scalar polynomial function defined in (2) and $\tilde{T}_i(t)$ is the ith entry of the vector $\mathbf{T}_n(t)$ that is given by

$$\tilde{T}_i(t) = \sum_{j=1}^{n} \varphi_{i,j} a_j(t) + T_i^* \tag{14}$$

where $\varphi_{i,j}$ is the ith entry of the basis vector $\boldsymbol{\varphi}_j$, a_j is the jth POD coefficient and T_i^* is the ith entry of $\mathbf{T}^*$ which contains the initial temperature profile of the bar. If we replace (14) in (13), then we obtain the component function F_i in terms of the POD coefficients $a_j(t)\ \forall j = 1, \ldots, n$. Notice that this function is a multivariate polynomial of degree $d_\mathrm{p} = 4$ in $a_j(t)\ \forall j = 1, \ldots, n$. Since $f_m(\mathbf{a}(t))$ (see (12)) is a linear combination of the functions $F_i(\mathbf{a}(t))$ $\forall i = 1, \ldots, N-1$, then $f_m(\mathbf{a}(t))$ is also a multivariate polynomial of degree $d_\mathrm{p} = 4$ in $a_j(t)\ \forall j = 1, \ldots, n$. After carrying out the mentioned replacements and simplifying the resulting expression, we obtain the following general representation for $f_m(\mathbf{a}(t))$,

$$f_m(\mathbf{a}(t)) = w_{m,0} + w_{m,1}a_1(t) + \cdots + w_{m,n}a_n(t) + \quad (15)$$
$$+ w_{m,(n+1)}a_1^2(t) + w_{m,(n+2)}a_1a_2 + \cdots + w_{m,(r-1)}a_n^{d_\mathrm{p}}$$

where $w_{m,i}$ is the real coefficient of the ith monomial ($i = 0, 1, 2, \ldots, r-1$) of the function $f_m(\mathbf{a}(t))$, d_p is the largest possible degree for each monomial and r is the number of monomials which is given by the following formula [7]:

$$r = \sum_{j=0}^{d_\mathrm{p}} \left(\begin{array}{c} n+j-1 \\ j \end{array} \right) = \sum_{j=0}^{d_\mathrm{p}} \frac{(n+j-1)!}{j!\,(n-1)!}\,. \quad (16)$$

For this particular problem, the number of monomials per component function is equal to 210. The monomials (without coefficients) $a_{i_1}a_{i_2}\cdots a_{i_s}$ of degree s $(1 \leq s \leq d_\mathrm{p})$ in (15) are built from the set $\mathcal{C}$,

$$\mathcal{C} = \big\{ \langle i_1, i_2, \cdots, i_s \rangle : 1 \leq i_1, i_2, \ldots, i_s \in \mathbb{Z}^+ \leq n, \quad (17)$$
$$i_1 \leq i_2 \leq \cdots \leq i_s \big\}$$

which contains the possible combinations of the indexes $i_1, i_2, \ldots, i_s$. By using the general representation of $\mathbf{f}(\mathbf{a}(t))$ defined in (15) through its component functions, we can recast the POD model (11) as follows:

$$\dot{\mathbf{a}}(t) = \mathbf{f}\left(\mathbf{a}(t)\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \quad (18)$$
$$\mathbf{T}_n(t) = \boldsymbol{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*$$

with

$$\mathbf{f}\left(\mathbf{a}(t)\right) = \left[f_1\left(\mathbf{a}(t)\right), \ldots, f_m\left(\mathbf{a}(t)\right), \ldots, f_n\left(\mathbf{a}(t)\right)\right]^T$$
$$f_m(\mathbf{a}(t)) = w_{m,0} + w_{m,1}a_1(t) + \cdots + w_{m,n}a_n(t) +$$
$$+ w_{m,(n+1)}a_1^2(t) + w_{m,(n+2)}a_1a_2 + \cdots + w_{m,(r-1)}a_n^{d_\mathrm{p}}.$$

From now on, this compact representation of (11) will be referred to as Polynomial POD model or P-POD model. Note that now the evaluation of $\mathbf{f}$ does not require the use of the high-dimensional vector function $\mathbf{F}$ anymore, and consequently we can expect to evaluate $\mathbf{f}$ much faster.

In general, if the discretized system used to approximate the PDE or PDEs is input affine and its nonlinearity is of polynomial type, then the vector function $\mathbf{f}(\mathbf{a}(t))$ can be represented by multivariate polynomials in terms of the POD coefficients. Notice that we might generate polynomial POD models from discretized systems with non-polynomial nonlinearities, provided that these nonlinearities can be approximated by low degree polynomials.

A. Calculation of the coefficients of the polynomials

If we define a vector $\mathbf{y}(t) \in \mathbb{R}^n$ as follows:

$$\mathbf{y}(t) = \dot{\mathbf{a}}(t) - \tilde{\mathbf{B}}_1 d(t) - \tilde{\mathbf{B}}_2 u(t) \quad (19)$$

then (11a) can be cast as $\mathbf{y}(t) = \mathbf{f}(\mathbf{a}(t))$.

The coefficients of the polynomials $f_m(\mathbf{a}(t))\ \forall m = 1, \ldots, n$ in (18) are computed in such a way that the fitting to the data generated by (19) at the time instants $t = k\Delta t$, for $k = 0, 1, 2, \ldots, N_{\mathrm{d}-1}$,

$$\mathcal{Y}_{N_\mathrm{d}} = \{\mathbf{y}(0), \mathbf{y}(\Delta t), \ldots, \mathbf{y}(N_\mathrm{d}\Delta t)\}$$

is optimal in the least squares sense. The dataset $\mathcal{Y}_{N_\mathrm{d}}$ is built from the data gathered during the snapshots experiment (see Section III-A) where N_d samples of the temperature profile were collected. This dataset is constructed as follows. From the snapshots experiment we know the values of $u(t = k\Delta t)$ and $d(t = k\Delta t)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$, and from the SVD of the snapshot matrix we have the evolution of $\mathbf{a}(t = k\Delta t)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$ (obtained by taking the first $n = 6$ rows of $\boldsymbol{\Sigma}\boldsymbol{\Psi}^T$). By using (11a) and the values of $u(k)$, $d(k)$ and $\mathbf{a}(k)$ for $k = 0, 1, 2, \ldots, N_\mathrm{d} - 1$, we calculate $\dot{\mathbf{a}}(t = k\Delta t)$ and afterwards $\mathbf{y}(t = k\Delta t)$ by means of (19) for the same time instants.

B. Reduction of the number of monomials

From (16) it is clear that the number of monomials of the P-POD model can be very large. In fact, this number increases exponentially with the number of POD coefficients, compromising the computational gain of the P-POD model. We can tackle this situation by properly selecting a reduced set of monomials whose combination provides a good approximation of the vector function $\mathbf{f}(\mathbf{a}(t))$. To this end, in this section we propose the use of sequential feature selection methods [8]. Here we will use the term "features" for referring to as the "monomials" of $\mathbf{f}\left(\mathbf{a}(t)\right)$.

The sequential feature selection methods have basically two components: the objective function, called the criterion, which the method seeks to minimize over all feasible feature subsets, and the sequential search algorithm. The algorithm adds or removes features from a candidate subset while evaluating the criterion. In general, an exhaustive evaluation of all possible feature subsets is infeasible, since we have to deal with $2^r = 2^{210} = 1.64 \times 10^{63}$ candidates. This is why a suboptimal search strategy is necessary for directing the feature selection process as it explores the space of all possible combination of features.

In this study, we use as objective function or criterion the Sum Squared Error (SSE) between the dataset $\mathcal{Y}_{N_\mathrm{d}}$ and the predictions of the candidate subset, and as a search algorithm the Sequential Forward Selection (SFS) in which the features are sequentially added to an empty candidate subset until a predetermined number of features has been appended. We have chosen SFS because is one of the simplest and computationally less demanding algorithms, due to the fact that the criterion is evaluated over small

TABLE I

PERFORMANCE OF THE POD MODELS

Test	POD model			Neural-POD model			P-POD model			P-POD-RS model		
	G_d	G_s	$\Delta T_{\max}$	G_d	G_s	$\Delta T_{\max}$	G_d	G_s	$\Delta T_{\max}$	G_d	G_s	$\Delta T_{\max}$
1	0.97	2.02	0.423	9.66	8.153	0.689	5.61	5.00	0.423	12.89	8.22	0.551
2	0.97	3.61	0.279	9.32	11.69	0.278	5.47	8.80	0.279	12.88	12.48	0.275
3	0.97	4.52	0.0518	9.32	14.03	0.0519	5.48	9.83	0.0518	12.91	13.85	0.0518
4	0.97	3.52	0.0594	9.30	11.43	0.0593	5.48	8.34	0.0594	12.92	12.72	0.0594

G_d is the computational gain in the calculation of the derivatives.
G_s is the computational gain in the simulation of the model.
$\Delta T_{\max}$ is the largest temperature deviation (error) in degrees celsius of the POD model.

subsets of monomials. For the evaluation of each candidate feature subset, a 10-fold cross-validation scheme is used and the coefficients of the different monomials are computed via least squares.

After running the feature selection algorithm, 25 features or monomials were finally chosen (SSE = $1.3727 \cdot 10^{-4}$). The polynomial POD model with a reduced set of monomials, which will be referred to as P-POD-RS model, has the following form,

$$\dot{\mathbf{a}}(t) = \mathbf{f}^{\mathcal{S}}\left(\mathbf{a}(t)\right) + \tilde{\mathbf{B}}_1 d(t) + \tilde{\mathbf{B}}_2 u(t) \qquad (20)$$

$$\mathbf{T}_n(t) = \mathbf{\Phi}_n \mathbf{a}(t) + \mathbf{T}^*$$

where $\mathbf{f}^{\mathcal{S}}(\mathbf{a}(t))$ is the vector function that approximates $\mathbf{f}(\mathbf{a}(t))$ and whose component functions contain the 25 monomials selected by the feature selection algorithm.

V. VALIDATION AND SIMULATION RESULTS

In this section we will validate and compare the reduced order models of the bar that were presented in Sections III, and IV, as well as the POD model based on an MLP proposed in [4] which will be referred to as Neural-POD model. To this end, the following tests were carried out:

- Test 1: A step of magnitude $1200 \cdot 10^3$ W$\cdot$m^{-3} is applied to $u(t)$ and a step of magnitude $500 \cdot 10^3$ W $\cdot$ m^{-3} is applied to $d(t)$.
- Test 2: Steps of magnitude $-1100 \cdot 10^3$ W $\cdot$ m^{-3} and $-400 \cdot 10^3$ W $\cdot$ m^{-3} are applied to $u(t)$ and $d(t)$ respectively.
- Test 3: A step of magnitude $500 \cdot 10^3$ W$\cdot$m^{-3} is applied to $u(t)$ and a step of magnitude $-200 \cdot 10^3$ W $\cdot$ m^{-3} is applied to $d(t)$.
- Test 4: Steps of magnitude $-400 \cdot 10^3$ W$\cdot$m^{-3} and $300 \cdot 10^3$ W $\cdot$ m^{-3} are applied to $u(t)$ and $d(t)$ respectively.

Given that the discretized model of the bar (4) consists of a set of stiff differential equations, an ODE solver that can deal with this condition has to be used. Hence, we used the function *ode23tb* of Matlab which implements TR-BDF2, an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two [9]. Along this work, not only (4) was solved with *ode23tb*, but also the ODEs describing each of the reduced order models. The solver was configured with a variable integration step and with a relative tolerance of 10^{-5} in all the cases. The initial conditions for the POD models were given by $\mathbf{a}(0) = \mathbf{\Phi}_n^T \mathbf{T}_n^{\Delta}(0) = \mathbf{0}$.

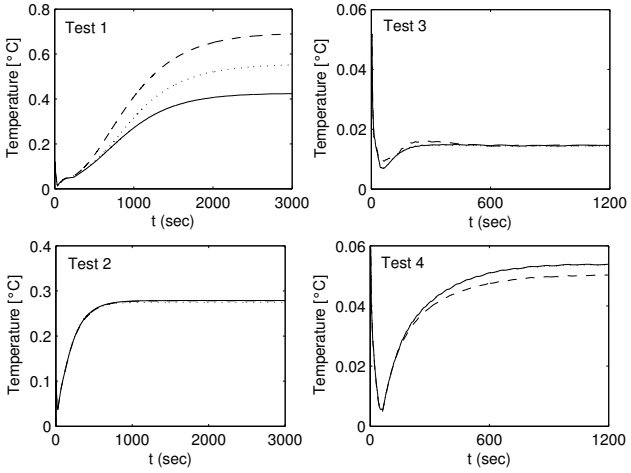

Fig. 2. Maximum temperature deviation of the POD models with respect to the full order model along the tests. Solid line - POD model and P-POD model. Dashed line - Neural-POD model. Dotted line - P-POD model with a reduced set of monomials.

In Table I we present the computational gain of the POD models with respect to the full order model (4), and we also include a measure of their accuracy. In this table, $\Delta T_{\max}$ is the largest temperature deviation (error) of the POD models regarding the high-dimensional model (4) along the entire test, and G_{d} and G_{s} quantify the computational gain of the POD models with respect to the full order model. They are defined as follows:

$$G_{\mathrm{d}} = \tilde{t}_{\mathrm{fom}} / \tilde{t}_{\mathrm{pod}}, \quad G_{\mathrm{s}} = t_{\mathrm{fom}} / t_{\mathrm{pod}}$$

where t_{fom} and t_{pod} are the times spent by the ODE solver for simulating the full order model and the POD model respectively, and $\tilde{t}_{\mathrm{fom}}$ and $\tilde{t}_{\mathrm{pod}}$ are the average times for calculating the derivatives along the test of the full order model and the POD model in that order. The values of G_{d} and G_{s} in Table I are average values found after 15000 and 1500 runs respectively. All the simulations were carried out in a PC with a dual core of 3 Ghz and a RAM memory of 2 GB.

Figure 2 shows the maximum temperature deviation of the different POD models with respect to the full order model along the tests, and Figure 3 depicts the evolution of the

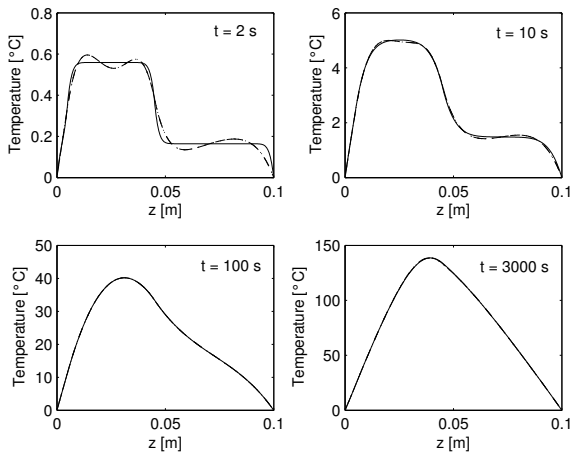

Fig. 3. Temperature profile at different time steps during Test 1. Solid line - Full order model (4). Dashed line - POD model and P-POD model. Dotted line - Neural-POD model. Dash-dotted line - P-POD-RS model. The temperatures are relative to the ambient temperature ($25°$C).

temperature profile of the bar during Test 1, the most severe test. In general, the accuracy of the POD models is good in spite of the big model-order reduction. Notice however that the Neural-POD model tends to be a little bit less accurate than the others. From Figure 3 we can observe how the temperature profiles of the POD models practically overlap each other.

From Table I we can notice that the derivatives of the P-POD model are calculated about 5.7 times faster than in the POD case. This computational gain has been achieved by exploiting the polynomial nature of the vector function $\mathbf{f}(\mathbf{a}(t))$ of (11) that allow us to write $\mathbf{f}(\mathbf{a}(t))$ in an efficient and compact form. This gain makes the simulation of the P-POD model about 2.4 times faster than the simulation of the POD model. Given that both models are equivalent, they have the same accuracy. Regarding the P-POD-RS model, the calculation of the derivatives is performed about 13.3 times faster than in the POD case and 2.3 times faster than in the P-POD case. This significant computational gain has been obtained by properly selecting a subset of monomials of the P-POD model. The impact of this gain on the simulation time is also remarkable. The simulation of P-POD-RS requires about 3.6 times less time than the simulation of the POD model. In addition, the difference in their responses is very small. The largest difference occurs during Test 1, and this difference is only of $0.13°$C.

VI. Conclusions

In this paper we have presented a way of accelerating the evaluation of nonlinear POD models derived from input-affine high-dimensional systems with polynomial nonlinearities. We have shown that by exploiting their polynomial nature, we can generate compact and efficient representations (P-POD models) which can be evaluated much faster than the POD models whose vector functions require the full spatial information of the original model. In addition, we have discussed how the sequential feature selection algorithms can help us to accelerate even more the evaluation of these P-POD models by selecting their most relevant monomials (suboptimal solution). A notable advantage of this approach with respect to the method based on neural networks is that we do not have the local minima problem, since we solve a convex optimization problem. However its drawback is that the number of monomials can be very large.

The simulation results showed that the P-POD model was as accurate as the POD model, but much faster. The P-POD model with a reduced set of monomials (P-POD-RS) provided a remarkable computational gain just by sacrificing a little bit its accuracy. This model together with the neural-POD model were the ones that offered the largest computational gains. Further research is necessary in order to evaluate the application of the approach proposed in this paper on more complex dynamic systems with harder nonlinearities.

VII. Acknowledgments

This research was supported by: • Research Council KUL: GOA MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; • Flemish Government: ○ FWO: PhD/postdoc grants, projects G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC); ○ IWT: PhD Grants, McKnow-E, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, POM; • Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); • EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, EMBOCOM; • Contract Research: AMINAL; • Other: ○ Helmholtz: viCERP; ○ ACCM; ○ Bauknecht; ○ Hoerbiger. Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium.

References

[1] P. Astrid, "Reduction of process simulation models: a proper orthogonal decomposition approach," Ph.D. dissertation, Technishche Universiteit Eindhoven, Eindhoven (Netherlands), November 2004.

[2] P. Astrid, S. Weiland, K. Willcox, and T. Backs, "Missing point estimation in models described by proper orthogonal decomposition," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2237–2251, November 2008.

[3] P. Astrid, "Fast reduced order modeling technique for large scale LTV systems," in *Proceedings of American Control Conference 2004*, vol. 1, 2004, pp. 762– 767.

[4] O. M. Agudelo, J. J. Espinosa, and B. DeMoor, "Acceleration of nonlinear POD models: a neural network approach," in *Proceedings of the European Control Conference 2009 (ECC'09)*, Budapest, Hungary, August 2009, pp. 1547–1552.

[5] A. Yousefi, B. Lohmann, J. Lienemann, and J. G. Korvink, "Nonlinear heat tranfer modelling and reduction," in *Proceedings of the 12th IEEE Mediterranean Conference on Control and Automation (MED '04)*, Kusadasi, Turkey, June 2004.

[6] L. Huisman, "Control of glass melting processes based on reduced CFD models," Ph.D. dissertation, Technishche Universiteit Eindhoven, Eindhoven (Netherlands), March 2005.

[7] M. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. Wiley–IEEE press, April 2003.

[8] R. Kohavi and H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997.

[9] L. F. Shampine and M. E. Hosea, "Analysis and implementation of TR–BDF2," *Applied Numerical Mathematics*, vol. 20, pp. 21–37, 1996.