


RESEARCH ARTICLE

WILEY

Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures

Stefania Fresca¹ | Giorgio Gobat² | Patrick Fedeli³ | Attilio Frangi² |
Andrea Manzoni¹ 

¹MOX - Department of Mathematics,
Politecnico di Milano, Milan, Italy

²Department of Civil and Environmental
Engineering, Politecnico di Milano, Milan,
Italy

³Analog and MEMS Group,
STMicroelectronics, Cornaredo, Italy

Correspondence

Andrea Manzoni, MOX - Department of
Mathematics, Politecnico di Milano, P.za
Leonardo da Vinci 32, 20133 Milano, Italy.
Email: andrea1.manzoni@polimi.it

Funding information

Fondazione Cariplo, Grant/Award
Number: 2019-4608

Abstract

We propose a non-intrusive deep learning-based reduced order model (DL-ROM) capable of capturing the complex dynamics of mechanical systems showing inertia and geometric nonlinearities. In the first phase, a limited number of high fidelity snapshots are used to generate a POD-Galerkin ROM which is subsequently exploited to generate the data, covering the whole parameter range, used in the training phase of the DL-ROM. A convolutional autoencoder is employed to map the system response onto a low-dimensional representation and, in parallel, to model the reduced nonlinear trial manifold. The system dynamics on the manifold is described by means of a deep feedforward neural network that is trained together with the autoencoder. The strategy is benchmarked against high fidelity solutions on a clamped-clamped beam and on a real micromirror with softening response and multiplicity of solutions. By comparing the different computational costs, we discuss the impressive gain in performance and show that the DL-ROM truly represents a real-time tool which can be profitably and efficiently employed in complex system-level simulation procedures for design and optimization purposes.

KEYWORDS

deep learning, MEMS, nonlinear dynamics, reduced order models

1 | INTRODUCTION

Nonlinear modeling in solid and structural mechanics has received an impressive boost in recent years thanks to the increasing availability of computational resources. In particular, the nonlinear dynamics of micro-electro-mechanical-systems (MEMS) has attracted great attention as nonlinearities-related effects play an important role at the microscale and they intrinsically arise during the regular functioning of MEMS. Although usually avoided and considered a nuisance, nonlinear signatures can be efficiently exploited paving the way to extraordinary performances for both existing devices and new nonlinear-ameliorated applications/working principles. Seminal works have already demonstrated the potential of devices that exploit nonlinear phenomena in microstructures. For instance, nonlinearities improve the frequency stability over temperature fluctuations of resonant accelerometers¹ and of resonators designed for clock applications;² the combination of mechanical (hardening) and electrostatic (softening) nonlinearities

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

also widens the linear response of MEMS sensors.^{3,4} Internal resonance, frequently observed in nano-micro mechanical resonators, has been employed to stabilize the oscillation frequency of self-sustaining resonators,⁵ and to improve the performances of Coriolis gyroscopes.⁶ Parametric resonance breaks into the engineering of filters,⁷ flow sensors,⁸ and mass sensors,⁹ as well as in boosting the sensitivity of gyroscopes.^{10,11} The design of MEMS devices based on the co-existence of multiple stable solutions has inspired gas sensors,¹² shock sensors,¹³ accelerometers,¹⁴ and flow sensors.¹⁵ In order to properly master and exploit nonlinearities, improved modeling capabilities are essential.

In this article, we focus on the large family of MEMS resonators (e.g., accelerometers, gyroscopes, magnetometers, micromirrors, time-keeping devices) which are primarily defined by the steady-state periodic response. Selected output quantities of interest like, for example, the maximum midspan deflection of a beam, or the rotation amplitude of a micromirror, should be predicted as a function of the actuation intensity and frequency—thus playing the role of input parameters—and ultimately yielding the so-called frequency response function (FRF). Moreover, the actuation can be electrostatic, piezoelectric, or magnetic, according to the considered applications, hence introducing additional sources of nonlinearities. Tracing the FRF thus requires to solve the problem multiple times (ranging from $O(10^3)$ to $O(10^7)$, depending upon the application at hand) for different values of a set of input parameters.

Even if numerical methods are emerging as a general solution to tackle the aforementioned tasks, the computational cost of high-fidelity approaches still remains a major issue, if not a true bottleneck. For instance, dedicated Harmonic Balance (HB) techniques or shooting procedures are overwhelmingly complex and time consuming.^{16,17} On the other hand, because of the low dissipation in vacuum packaged devices, ultimately leading to long transients, time marching schemes are hardly computationally affordable, yielding to high-dimensional, nonlinear dynamical systems to be solved. The community of nonlinear dynamics at the microscale therefore lacks of approaches that are suitable for the new generation of MEMS, in specific scenarios where multi-queries and real-time performances are required for the sake of design, testing, and optimization. In all these cases, relying on high-fidelity computational techniques is indeed unaffordable.

Among different options, reduced order models (ROMs) represent a key numerical tool in order to generate efficient, yet reliable, approximations to the solution of parametrized differential problems. In particular, projection-based ROMs reshape the original high-fidelity problem into a nonlinear, dynamical system featuring a much lower dimension, yet capable to capture the physical features of the problem at hand.¹⁸⁻²⁰ However, projection-based ROMs are intrusive techniques, possibly expensive to be assembled in the case of high order polynomial (or nonpolynomial) nonlinearities, and hurdles to scale over many parameters instances.²¹⁻²³

In this framework, extensions of the modal methods available for linear problems to the nonlinear regime are receiving increasing attention.²⁴ Nevertheless, the computation of the invariant manifolds of nonlinear normal modes starting from the normal form theory has been developed to maturity only for relatively small-scale systems²⁵ and applications to large scale problems are still the object of intensive research.²⁶ Thus, the projection of dynamical systems onto low-dimensional nonlinear manifolds is still an open challenge. Data-driven approaches based on proper orthogonal decomposition (POD) have been recently successfully applied to MEMS by Gobat et al.,²⁷ however, these are still linear approaches that might experience some difficulties in reproducing the correct curvature of invariant manifolds while retaining only few DOFs in the ROM and eventually fail to reach real-time performances.

Exploiting machine learning methods for constructing surrogates of dynamical systems has recently been an area of increasing interest for the system dynamics community. Among others, huge success has been encountered by the physics informed neural networks (PINN)²⁸ which have been applied in multiple contexts and also in solid mechanics.²⁹ These neural networks leverage the automatic differentiation feature to directly enforce partial differential equations. However, so far they still do not serve the aim of generating ROMs for parameter dependent problems.

Deep learning (DL) techniques come as an inspiration to handle the complex reduction process of dynamical systems, unveiling low-dimensional features from black-box data streams.³⁰⁻³² Brunton and coworkers recently applied the SINDy method³³⁻³⁵ in combination with autoencoder neural networks to discover the underlying model of dynamical systems.³⁶ Fresca et al.,^{37,38} proposed a non-intrusive DL-based ROM technique, which we refer to as deep learning-based reduced order model (DL-ROM). Combining in a suitable way a convolutional autoencoder (AE) and a deep feedforward neural network (DFNN), the DL-ROM technique enables the construction of an efficient ROM, whose dimension is as close as possible to the number of parameters upon which the solution of the differential problem depends. The encoder part performs an operation of feature extraction forcing the high dimensional data to be reduced, at the bottleneck layer, to few latent variables. We highlight that this approach builds on the idea that the system dynamics develops on a very low dimensional curved invariant manifold. Indeed this is very often the case for real MEMS devices which, in the linear regime, oscillate according to a specific mode of interest, while this turns into the corresponding nonlinear normal mode (NNM) as the actuation intensity increases.^{24,26} The procedure exploits snapshots taken from a set of

FOM solutions (for selected parameter values and time instances) and deep neural network architectures to learn, in a non-intrusive way (i) the nonlinear trial manifold where the ROM solution is sought (ii) the nonlinear reduced dynamics, and (iii) and the reconstruction of the approximate FOM response starting from the latent variables. In the framework proposed by Fresca et al.,³⁸ a further dimensionality reduction carried out on the FOM data through POD, yielding the so-called POD DL-ROM technique, also allows to speedup training times and to compress data dimensions, enhancing the construction of DL-ROMs. Neural networks have also been used to model simple structures by Simpson et al.,³⁹ with two main differences: the convolutional autoencoder uses long short-term memory (LSTM) networks; a statistical regression model is used instead of the DFNN. The adoption of LSTM cells is beneficial for the prediction of the time evolution beyond the training window, which is not targeted in the present contribution focusing on periodic responses. More in general, DL algorithms in conjunction with POD have already been exploited to address long-term predictions in time, however without including parameter dependencies.^{40,41} Recurrent neural networks have been considered by Gonzalez and Balajewicz⁴² to evolve low-dimensional states of unsteady flows, exploiting either POD or a convolutional recurrent autoencoder to extract low-dimensional features from snapshots. DL algorithms have also been used to describe the reduced trial manifold where the approximation is sought, then exploiting a minimum residual formulation to derive the ROM as done by Lee and Carlberg.³²

In this article we propose a new version of the DL-ROM discussed above, in which a first dimensionality reduction of the data is achieved by means of a POD-Galerkin (POD-G) ROM making use of a single global set of POD modes. It worth stressing that that this simple technique might fail in complex scenarios like, for example, fluid structure simulations for flutter analysis. For these applications sophisticated interpolation algorithms, as proposed for example, in References 43–45, might reveal necessary. However, here we restrain ourselves to the specific class of nonlinear vibrations of microstructures undergoing non-infinitesimal, but still moderate transformations. This context, though limited in scope, has a huge practical importance given the spread of MEMS in everyday life. The nonlinearities involved are known as geometrical and the reduced models of these structures are generalized Duffing equations.²⁶ The accuracy of the POD-Galerkin ROM with a single global set of POD modes has been extensively investigated for these applications by Gobat et al.²⁷ showing very good predictive capabilities when validated against the original high-fidelity FOM. However, for the sake of completeness, also in the present contribution the error of POD-G with respect to the FM will always be presented. Before continuing, we stress anyway that the data generation phase is not the main focus of this contribution, which rather aims at investigating the ability of the DL approach to identify the main features of the response and help achieve a model order reduction. Different problems will indeed require different data generation strategies. Moreover, the speed-up performance achieved at testing time is due to the use of the DL-ROM and is not an attribute of the data generation strategy used only during the training process.

In our case, selecting the POD-G, the costly FOM data generation phase is greatly reduced and the training of the DL-ROM is performed using cheaper POD-G snapshots covering the whole parameter range. This allows addressing real industrial examples in complex scenarios where multiplicity of solutions exist thus requiring specific provisions.

Compared to other surrogate models exploiting machine/deep learning algorithms, a distinguishing feature of POD-G DL-ROMs is their capability to compute the whole solution field, for any new parameter instance and time instant, at testing time, thus enabling the extremely efficient evaluation of any output quantity of interest depending on the solution field. The technique proposed in this work can be considered as the data driven counterpart of the direct normal form method for invariant manifolds that has been recently applied to large scale finite element systems of mechanical structures.^{26,46,47}

The structure of the article is as follows. After a brief definition in Section 2 of the problem at hand and of the type of nonlinearities involved, we discuss the proposed data-driven procedure in Section 3. The POD-G ROM, that is, the first level of dimensionality reduction, is presented in Section 3.1, while the second deep learning level is analyzed in Section 3.2, detailing the structure of the autoencoder and of the FF neural network. Two applications are then presented in Section 4, starting with the academic example of a clamped-clamped beam in Section 4.2, and later addressing a real micromirror in Section 4.3.

2 | PROBLEM FORMULATION

We will consider MEMS devices with given fixed geometry subjected to a periodic forcing of angular frequency ω and modulated in amplitude by the coefficient β . The input parameter vector is $\mu = [\omega, \beta] \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$, with \mathcal{P} a closed and bounded set of dimension $n_\mu = 2$. These devices are usually characterized by several FRFs. For instance, in the specific case of

the doubly clamped beam analyzed in Section 4.2, the FRF plotted in Figure 5F expresses the mid-span displacement u_{mid} of the beam in terms of $\boldsymbol{\mu}$. Each dot represents the maximum value of u_{mid} in the corresponding periodic solution. Moreover one might desire to monitor the evolution of stresses or strains at other specific points in order to guarantee a correct operation of the device. Providing all these output data is the main goal of any numerical technique. Therefore we aim to formulate a fast and reliable *reduced-order* technique capable of predicting periodic responses of period T , with $T = 2\pi/\omega$, and of providing as output the corresponding full field response in terms of displacements, strains and stresses.

Many microsystems, like for example, micromirrors, undergo large transformations and in particular large rotations. The device initially occupies the domain Ω_0 described by material coordinates \mathbf{X} and is subjected to the transformation $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t)$, with \mathbf{u} displacement vector and \mathbf{x} actual spatial coordinates. The boundary $\partial\Omega_0$ is partitioned in $\partial\Omega_D$ and $\partial\Omega_N$ where Dirichlet and Neumann periodic boundary conditions are enforced, respectively. In this contribution, for the sake of simplicity, only homogeneous boundary conditions will be considered. By assumption, this partition does not depend on the transformation thus ruling out specific applications including, for example, evolving contact. The device is actuated by given time-periodic body forces $\mathbf{B}(\mathbf{X}, t)$. The overall governing system of PDEs with periodic time boundary conditions reads:

$$\rho_0 \ddot{\mathbf{u}}(\mathbf{X}, t) - \nabla \cdot \mathbf{P}(\mathbf{X}, t) - \rho_0 \mathbf{B}(\mathbf{X}, t; \boldsymbol{\mu}) = \mathbf{0} \quad \text{for } (\mathbf{X}, t) \text{ in } \Omega_0 \times [0, T], \quad (1a)$$

$$\mathbf{P}(\mathbf{X}, t) \cdot \mathbf{N}(\mathbf{X}) = \mathbf{0} \quad \text{for } (\mathbf{X}, t) \text{ in } \partial\Omega_N \times [0, T], \quad (1b)$$

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{0} \quad \text{for } (\mathbf{X}, t) \text{ in } \partial\Omega_D \times [0, T], \quad (1c)$$

$$\mathbf{u}(\mathbf{X}, 0) = \mathbf{u}(\mathbf{X}, T) \quad \text{for } \mathbf{X} \text{ in } \Omega_0, \quad (1d)$$

$$\dot{\mathbf{u}}(\mathbf{X}, 0) = \dot{\mathbf{u}}(\mathbf{X}, T) \quad \text{for } \mathbf{X} \text{ in } \Omega_0, \quad (1e)$$

$$\mathbf{S}(\mathbf{X}, t) = \mathcal{A}(\mathbf{X}) : \mathbf{E}(\mathbf{X}, t) \quad \text{for } (\mathbf{X}, t) \text{ in } \Omega_0 \times [0, T], \quad (1f)$$

$$\mathbf{E}(\mathbf{X}, t) = \frac{1}{2} (\nabla \mathbf{u}(\mathbf{X}, t) + \nabla^T \mathbf{u}(\mathbf{X}, t) + \nabla^T \mathbf{u}(\mathbf{X}, t) \cdot \nabla \mathbf{u}(\mathbf{X}, t)) \quad \text{for } (\mathbf{X}, t) \text{ in } \Omega_0 \times [0, T], \quad (1g)$$

$$\mathbf{P}(\mathbf{X}, t) = (\mathbf{1} + \nabla \mathbf{u}(\mathbf{X}, t)) \cdot \mathbf{S}(\mathbf{X}, t), \quad \text{for } (\mathbf{X}, t) \text{ in } \Omega_0 \times [0, T]. \quad (1h)$$

Equation (1a) expresses the conservation of momentum where ρ_0 is the initial density and \mathbf{P} is the first Piola-Kirchhoff stress.⁴⁸ Equations (1b) and (1c) are the Neumann and Dirichlet boundary conditions respectively. Equations (1d) and (1e) enforce the periodicity condition on displacements and velocities. The device is made of cubic single crystal silicon or polysilicon, thus admitting only small strains, a condition which is well described by the Saint Venant-Kirchhoff constitutive model Equation (1f) between the Second Piola-Kirchhoff stress \mathbf{S} and the Green-Lagrange strain tensor \mathbf{E} (Equation 1g) through the fourth-order elasticity tensor \mathcal{A} endowed with major and minor symmetries. Finally, Equation (1h) formulates the link between the two Piola-Kirchhoff stress tensors.⁴⁸

The weak form of the momentum conservation, that is, Equations (1a) and (1b), reads:

$$\int_{\Omega_0} \rho_0 \ddot{\mathbf{u}}(\mathbf{X}, t) \cdot \mathbf{w}(\mathbf{X}) \, d\Omega_0 + \int_{\Omega_0} \mathbf{P}(\mathbf{X}, t) : \nabla^T \mathbf{w}(\mathbf{X}) \, d\Omega_0 = \int_{\Omega_0} \rho_0 \mathbf{B}(\mathbf{X}, t; \boldsymbol{\mu}) \cdot \mathbf{w}(\mathbf{X}) \, d\Omega_0 \quad \forall \mathbf{w} \in H_0^1(\Omega_0), \quad (2)$$

where \mathbf{w} is the test velocity selected in $H_0^1(\Omega_0)$, that is, the space of functions with finite energy that vanish on the portion $\partial\Omega_D \subset \partial\Omega_0$ where Dirichlet boundary conditions are prescribed. Within the present context, it is worth stressing that Equation (2) exactly accounts for geometric (elastic and inertia) nonlinearities, for example, large rotations or nonlinear mode coupling.

The spatial discretization of Equation (2), for example, by means of the finite element method, with the additional inclusion of a Rayleigh model damping term, yields a system of coupled nonlinear differential equations representing the full order model (FOM):

$$\mathbf{M} \ddot{\mathbf{u}}_h(t) + \mathbf{C} \dot{\mathbf{u}}_h(t) + \mathbf{K} \mathbf{u}_h(t) + \mathbf{G}(\mathbf{u}_h, \mathbf{u}_h) + \mathbf{H}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) = \mathbf{F}(t; \boldsymbol{\mu}), \quad t \in (0, T), \quad (3a)$$

$$\mathbf{u}_h(0) = \mathbf{u}_h(T), \quad \dot{\mathbf{u}}_h(0) = \dot{\mathbf{u}}_h(T), \quad (3b)$$

where the vector $\mathbf{u}_h(t) \in \mathbb{R}^{N_h}$ collects the N_h unknown displacements nodal values, $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ is the mass matrix, $\mathbf{C} = (\omega_0/Q)\mathbf{M}$ is the Rayleigh mass-proportional damping matrix—considering a reference eigenfrequency ω_0 and a quality factor Q —and $\mathbf{F}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the nodal force vector which depends on the vector of parameters $\boldsymbol{\mu}$. The internal force vector has been exactly decomposed in linear, quadratic, and cubic power terms of the displacement: $\mathbf{K} \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix related to the linearized system, while $\mathbf{G} \in \mathbb{R}^{N_h}$ and $\mathbf{H} \in \mathbb{R}^{N_h}$ are vectors given by monomials of second and third order, respectively. We stress that the components of these vectors can be expressed using an indicial notation as

$$G_i = \sum_{j,k=1}^{N_h} G_{ijk} u_{h,j} u_{h,k}, \quad H_i = \sum_{j,k,l=1}^{N_h} H_{ijkl} u_{h,j} u_{h,k} u_{h,l}, \quad i = 1, \dots, N_h.$$

Equation (3) represents our high-fidelity FOM which depends on the input parameters μ . Our goal is the efficient numerical approximation of the solution manifold:

$$\mathcal{S} = \{ \mathbf{u}_h(t; \mu) \mid t \in [0, T) \text{ and } \mu \in \mathcal{P} \subset \mathbb{R}^{n_\mu} \} \subset \mathbb{R}^{N_h}. \quad (4)$$

In particular, assuming that, for any given parameter $\mu \in \mathcal{P}$, the FOM admits a unique solution for each $t \in (0, T)$, that is, the dynamical system Equation (3) has a unique trajectory for each parameter instance, the intrinsic dimension of the solution manifold is at most $n_\mu + 1 \ll N_h$, where n_μ is the number of parameters (time thus plays the role of an additional coordinate).

The numerical solution of the FOM Equation (3) to compute the steady state response is a challenge in itself for large scale problems. One option is the use of time marching methods to simulate a sufficiently large number N_c of cycles, where N_c is typically inversely proportional to the damping. This technique resorts to robust algorithms implemented in most of the commercial software, but when damping is very small, as in most MEMS devices, the computational effort may not be affordable. In other approaches, like the HB method,^{49,50} the unknown displacements are expressed as the sum of Fourier components thus automatically respecting periodicity conditions. However, their implementation requires dedicated codes and non-standard computing facilities.

3 | DATA-DRIVEN MODEL ORDER REDUCTION

In this section, we briefly review the construction of POD-Galerkin (POD-G) ROMs and of the POD-G deep learning-based ROM (POD-G DL-ROM) technique which combines the best features of POD and DL algorithms in order to achieve real-time simulations of the structural behavior of MEMS devices.

3.1 | POD-Galerkin ROM (POD-G ROM)

A simple way to build a ROM able to approximate Equation (4) is through the introduction of a linear reduced trial manifold $\tilde{\mathcal{S}}_h^{N,lin} = \text{Col}(\mathbf{V})$ of dimension $N \ll N_h$, spanned by the N columns of a matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$. Thus we can achieve an approximation $\tilde{\mathbf{u}}_h(t, \mu) \approx \mathbf{u}_h(t, \mu)$ as:

$$\tilde{\mathbf{u}}_h(t, \mu) = \mathbf{V} \mathbf{u}_N(t; \mu), \quad (5)$$

in which $\tilde{\mathbf{u}}_h : [0, T) \times \mathcal{P} \rightarrow \tilde{\mathcal{S}}_h$ and $\mathbf{u}_N(t; \mu) \in \mathbb{R}^N$, $\forall t \in [0, T)$ and $\mu \in \mathcal{P}$, is the vector of degrees of freedom (DOFs) of the ROM approximation. The most commonly used technique to build the subspace $\tilde{\mathcal{S}}_h^N$ bases is through POD. The first step in the construction of a POD-G ROM requires to generate a matrix $\mathbf{S}_u \in \mathbb{R}^{N_h \times N_s}$, whose N_s columns collect snapshots of the FOM solutions, obtained for different values of the parameters μ

$$\mathbf{S}_u = [\mathbf{u}_N(t^1, ; \mu_1) | \dots | \mathbf{u}_N(t^{N_t}, ; \mu_1) | \dots | \mathbf{u}_N(t^1, ; \mu_{N_{\text{train}}}) | \dots | \mathbf{u}_N(t^{N_t}, ; \mu_{N_{\text{train}}})]. \quad (6)$$

Each snapshot in \mathbf{S}_u is sampled inside a certain time interval $[0, T]$ partitioned in N_t time steps, with $t^k, k = 1, \dots, N_t$, and the parameter space \mathcal{P} where a set of N_{train} parameter instances $\mu_k, k = 1, \dots, N_{\text{train}}$, is selected. Next, POD computes the singular value decomposition (SVD) of the matrix \mathbf{S}_u

$$\mathbf{S}_u = \mathbf{U} \mathbf{\Sigma} \mathbf{Z}^T,$$

where the columns of the orthonormal matrix $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$ are the left singular vectors and the columns of the orthonormal matrix $\mathbf{Z} \in \mathbb{R}^{N_s \times N_s}$ are the right singular vectors. The diagonal elements of $\mathbf{\Sigma} \in \mathbb{R}^{N_h \times N_s}$ are the singular values of the matrix \mathbf{S}_u and are conventionally ordered from the largest to the smallest. In particular, POD selects

the columns of \mathbf{V} in Equation (5) as the first N left singular vectors of \mathbf{S}_u , often called POD modes (POMs) in the literature.^{20,51,52}

Once the linear trial POD subspace has been obtained, projecting the FOM Equation (3) onto the POD subspace yields the structural dynamics geometric POD-G ROM, under the form of a N -dimensional nonlinear ODE system, whose solution provides the dynamics of the generalized coordinates $\mathbf{u}_N \in \mathbb{R}^N$:

$$\mathbf{M}^{\text{POD}} \ddot{\mathbf{u}}_N + \mathbf{C}^{\text{POD}} \dot{\mathbf{u}}_N + \mathbf{K}^{\text{POD}} \mathbf{u}_N + \mathbf{G}^{\text{POD}}(\mathbf{u}_N, \mathbf{u}_N) + \mathbf{H}^{\text{POD}}(\mathbf{u}_N, \mathbf{u}_N, \mathbf{u}_N) = \mathbf{F}^{\text{POD}}(\mathbf{u}_N, t; \boldsymbol{\mu}), \quad t \in (0, T), \quad (7)$$

where

$$\begin{aligned} \mathbf{M}^{\text{POD}} &= \mathbf{V}^T \mathbf{M} \mathbf{V}, \quad \mathbf{C}^{\text{POD}} = \mathbf{V}^T \mathbf{C} \mathbf{V}, \quad \mathbf{K}^{\text{POD}} = \mathbf{V}^T \mathbf{K} \mathbf{V}, \\ \mathbf{F}^{\text{POD}} &= \mathbf{V}^T \mathbf{F}, \quad G_i^{\text{POD}} = g_{ijk}^{\text{POD}} u_{N,j} u_{N,k}, \quad H_i^{\text{POD}} = h_{ijkl}^{\text{POD}} u_{N,j} u_{N,k} u_{N,l}, \end{aligned}$$

with $\mathbf{M}^{\text{POD}}, \mathbf{C}^{\text{POD}}, \mathbf{K}^{\text{POD}} \in \mathbb{R}^N \times \mathbb{R}^N$. The computation of the vectors \mathbf{G}^{POD} and \mathbf{H}^{POD} entails $O(N^3)$ and $O(N^4)$ terms, respectively. Note that the coefficients g_{ijk}^{POD} and h_{ijkl}^{POD} can be precomputed, and that the reduced problem can be assembled efficiently thanks to its polynomial nature, thus avoiding the use of hyper-reduction techniques such as the (discrete) empirical interpolation method.⁵³⁻⁵⁵

The POD-G approach has been recently benchmarked on several MEMS including beams, arches, and mirrors. With reference to MEMS structures Gobat et al.²⁷ have shown that, provided that a sufficient number of POD modes is included in the trial space, the technique accurately reproduces the response of the device. However, in particular when large rotations are involved, the dimension of the trial space increases and solution of the ROM with continuation techniques comes with a high computational cost, failing to serve the final goal of generating a real-time simulation tool. As recalled in the introduction, the approach however delivers a remarkable accuracy despite using a single global basis, and does not require elaborate strategies like those proposed for example, in References 43-45 for different applications. This very good performance of the POD-G approach is strongly linked to the fact that the range of variation of the frequency parameter, the one which impacts the most on the model, is actually very limited (as typical of MEMS resonators) and that the nonlinearities are mild, since the solution manifold is regular. All these factors usually play a remarkable role in view of making ROMs based on the linear superimposition of modes accurate and reliable.

3.2 | POD-Galerkin-enhanced deep learning based-reduced order models (POD-G DL-ROMs)

POD-DL-ROMs are non-intrusive ROMs, which aim at approximating the map $(t, \boldsymbol{\mu}) \rightarrow \mathbf{u}_h(t, \boldsymbol{\mu})$ by describing both the trial manifold and the reduced dynamics through deep neural networks.⁵⁶ To reduce the dimensionality of the snapshots and avoid feeding training data of very large dimension N_h , POD is first applied—realized through randomized SVD (rSVD)⁵⁷—to the snapshot set \mathbf{S}_u ; then, a DL-ROM is built to approximate the map between $(t, \boldsymbol{\mu})$ and the POD generalized coordinates $\mathbf{u}_N(t; \boldsymbol{\mu})$.

The POD-DL-ROM technique is extremely efficient and it is able to model highly nonlinear problems by identifying the manifold underlying the dynamics of the system in a complete data-driven and black-box, non-intrusive way. On the other hand, its data-driven nature implies that the sampled data provided must span the parameter space of interest and contain all the information necessary to accurately approximate the solution manifold. In this way, the size of the training dataset increases with the number of parameters considered in the FOM.

Instead, POD projection-based ROMs exploit a data-driven process to generate the basis of the linear subspace, later this basis is used to project the nonlinear system given by the spatial discretization. For instance (exact) projection procedures like the POD-G ROM outlined in Section 3.1, lead to a nonlinear system of ODEs whose DOFs can be physically interpreted and that can be solved with many different integration packages (i.e., Auto07p,⁵⁸ MANLAB,⁵⁹ Nlvib,⁶⁰ Matlab built-in functions, BifurcationKit⁶¹ etc.). On the other hand, such POD-G ROM is strongly intrusive and it is way less efficient than a DL-ROM, because the ODE system must be solved up to a given time instant of interest for a given parameter instance. These conditions may be acceptable when a limited number of inquiries are needed, but when a refined inspection of the system response dynamics is necessary for optimal design purpose or when a real-time solution is needed (e.g., online control) POD-G ROMs, and in general intrusive approaches, fail.

In this contribution, we exploit the most appealing features of both techniques to achieve a real-time simulation of a MEMS. A sketch of the proposed procedure, named POD-Galerkin-enhanced deep learning-based ROM (POD-G DL-ROM), is outlined in Figure 1.

First, a few FOM simulations, that loosely span the parameter space, are performed. The generated high-fidelity snapshots are processed with SVD and a subset of POD modes is selected. The number of POD modes kept in the subspace must be defined in order to guarantee a good approximation of the underlying manifold as shown by Gobat et al.,²⁷ where POD-G ROMs are applied to mechanical systems with low damping like MEMS. By exploiting the generated linear subspace, the POD-G ROM is built following the procedure detailed in Section 3.1. The POD-G ROM depends explicitly on the model parameters of interest, that is, the load amplitude β and the forcing frequency ω , and it is solved, at testing time, to create snapshots of the ROM intrinsic coordinates finely spanning the parameter space \mathcal{P} .

The resulting reduced solutions are provided as input to the POD-G DL-ROM neural network during the training stage. Following the DL-ROM technique outlined by Fresca et al.,³⁸ the DFNN is simultaneously trained to provide as output the minimal coordinate vector for the same parameter instance. Finally, the decoder function of the convolutional AE maps the minimal coordinates to the approximated intrinsic coordinates.

More precisely, the POD-G DL-ROM approximation of the FOM solution $\mathbf{u}_h(t; \mu)$ is

$$\tilde{\mathbf{u}}_h(t; \mu, \theta_{DF}, \theta_D) = \mathbf{V} \tilde{\mathbf{u}}_N(t; \mu, \theta_{DF}, \theta_D) \approx \mathbf{V} \mathbf{u}_N(t; \mu, \theta_{DF}, \theta_D) \approx \mathbf{u}_h(t; \mu),$$

that is, it is sought in a linear trial manifold of (potentially large) dimension N ,

$$\tilde{\mathcal{S}}_h^N = \{ \mathbf{V} \tilde{\mathbf{u}}_N(t; \mu, \theta_{DF}, \theta_D) \mid \tilde{\mathbf{u}}_N(t; \mu, \theta_{DF}, \theta_D) \in \mathbb{R}^N, t \in [0, T], \mu \in \mathcal{P} \} \subset \mathbb{R}^{N_h}, \quad (8)$$

by applying the DL-ROM strategy³⁸ to approximate $\mathbf{u}_N(t; \mu)$ —rather than directly $\mathbf{V}^T \mathbf{u}_h(t; \mu)$. The DL-ROM approximation $\tilde{\mathbf{u}}_N(t; \mu, \theta_{DF}, \theta_D) \approx \mathbf{u}_N(t; \mu)$ takes the form

$$\tilde{\mathbf{u}}_N(t; \mu, \theta_{DF}, \theta_D) = \mathbf{f}_N^D(\phi_n^{DF}(t; \mu, \theta_{DF}); \theta_D), \quad (9)$$

and is sought in a reduced nonlinear trial manifold $\tilde{\mathcal{S}}_N^n$ of very small dimension $n \ll N$; usually, $n \approx n_\mu + 1$ —here time is considered as an additional parameter. As in a POD-DL-ROM, in a POD-G DL-ROM both the reduced dynamics and the reduced nonlinear manifold (or trial manifold) where the ROM solution is sought must be learnt. In particular:

- *Reduced dynamics learning.* To describe the system dynamics on the nonlinear trial manifold $\tilde{\mathcal{S}}_N^n$, the intrinsic coordinates of the approximation $\tilde{\mathbf{u}}_N$ are defined as

$$\mathbf{u}_n(t; \mu) = \phi_n^{DF}(t; \mu, \theta_{DF}),$$

where $\phi_n^{DF}(\cdot; \cdot, \theta_{DF}) : [0, T] \times \mathbb{R}^{n_\mu} \rightarrow \mathbb{R}^n$ is a DFNN, consisting of the repeated composition of a nonlinear activation function, applied to a linear transformation of the input, multiple times. Here θ_{DF} denotes the DFNN parameters vector, collecting the weights and biases of each of its layers;

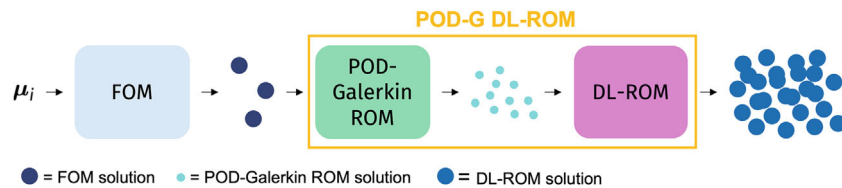


FIGURE 1 POD-G DL-ROM procedure scheme. By solving the FOM for few parameter instances, a small number of FOM snapshots are generated. The high-fidelity snapshots are used to compute the POD basis matrix \mathbf{V} and generate the POD-G ROM. This is later exploited to compute a large number of reduced snapshots, for different parameter instances, able to finely span the parameter space of interest. These snapshots represent the input to the POD-G DL-ROM neural network. Once the POD-G DL-ROM is trained, it is used to generate all the approximate solutions required.

- *Nonlinear trial manifold learning.* To model the reduced nonlinear trial manifold \tilde{S}_N^n , we employ the decoder function of a convolutional autoencoder (AE), that is,

$$\tilde{S}_N^n = \{\tilde{\mathbf{u}}_N(t; \boldsymbol{\mu}) = \mathbf{f}_N^D(\boldsymbol{\phi}_n^{DF}(t; \boldsymbol{\mu}, \boldsymbol{\theta}_{DF}); \boldsymbol{\theta}_D) \mid \boldsymbol{\phi}_n^{DF}(t; \boldsymbol{\mu}, \boldsymbol{\theta}_{DF}) \in \mathbb{R}^n, t \in [0, T], \boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_\mu}\} \subset \mathbb{R}^N, \quad (10)$$

where $\mathbf{f}_N^D(\cdot; \boldsymbol{\theta}_D) : \mathbb{R}^n \rightarrow \mathbb{R}^N$ denotes the decoder function of a convolutional AE obtained as the composition of several layers (some of which are convolutional), depending upon a vector $\boldsymbol{\theta}_D$ collecting all the corresponding weights and biases.

Finally, the encoder function $\mathbf{f}_n^E(\cdot; \boldsymbol{\theta}_E) : \mathbb{R}^N \rightarrow \mathbb{R}^n$ —depending upon a vector $\boldsymbol{\theta}_E$ of parameters—of the convolutional AE can be used to map the POD-G reduced solution (or approximated intrinsic coordinates) $\mathbf{u}_N(t, \boldsymbol{\mu})$ associated to $(t, \boldsymbol{\mu})$ onto a low-dimensional representation

$$\tilde{\mathbf{u}}_n(t; \boldsymbol{\mu}, \boldsymbol{\theta}_E) = \mathbf{f}_n^E(\mathbf{u}_N(t; \boldsymbol{\mu}); \boldsymbol{\theta}_E).$$

Hence, provided the parameter matrix $\mathbf{L} \in \mathbb{R}^{(n_\mu+1) \times N_s}$ defined as

$$\mathbf{L} = [(\mathbf{t}^1, \boldsymbol{\mu}_1) \mid \dots \mid (\mathbf{t}^{N_t}, \boldsymbol{\mu}_1) \mid \dots \mid (\mathbf{t}^1, \boldsymbol{\mu}_{N_{\text{train}}}) \mid \dots \mid (\mathbf{t}^{N_t}, \boldsymbol{\mu}_{N_{\text{train}}})], \quad (11)$$

and the snapshot matrix \mathbf{S}_u , training a POD-G DL-ROM requires to solve the optimization problem

$$\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{N_{\text{train}} N_t} \sum_{i=1}^{N_{\text{train}}} \sum_{k=1}^{N_t} \mathcal{L}(\mathbf{t}^k, \boldsymbol{\mu}_i; \boldsymbol{\theta}), \quad (12)$$

where the *per-example* loss function $\mathcal{L}(\mathbf{t}^k, \boldsymbol{\mu}_i; \boldsymbol{\theta})$ is given by the sum of two terms

$$\begin{aligned} \mathcal{L}(\mathbf{t}^k, \boldsymbol{\mu}_i; \boldsymbol{\theta}) &= \frac{\omega_h}{2} \|\mathbf{u}_N(\mathbf{t}^k; \boldsymbol{\mu}_i) - \tilde{\mathbf{u}}_N(\mathbf{t}^k; \boldsymbol{\mu}_i, \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D)\|^2 \\ &\quad + \frac{1 - \omega_h}{2} \|\tilde{\mathbf{u}}_n(\mathbf{t}^k; \boldsymbol{\mu}_i, \boldsymbol{\theta}_E) - \mathbf{u}_n(\mathbf{t}^k; \boldsymbol{\mu}_i, \boldsymbol{\theta}_{DF})\|^2, \end{aligned} \quad (13)$$

with $\boldsymbol{\theta} = (\boldsymbol{\theta}_E, \boldsymbol{\theta}_{DF}, \boldsymbol{\theta}_D)$. The former term in Equation (13) is the reconstruction error between the POD-G ROM reduced solutions and the POD-G DL-ROM approximations while the latter is the misfit between the intrinsic coordinates and the output of the encoder function. Finally, $\omega_h \in [0, 1]$ is a prescribed weighting parameter. The architecture of the POD-G DL-ROM neural network is shown in Figure 2; note that, at testing time, we can discard the encoder function of the convolutional AE.

We remark that the computation of the low-fidelity snapshots, obtained as reduced solution of a POD-G ROM for several parameter instances, can be considered as an extension of the idea at the basis of the POD-DL-ROM technique, where the high-fidelity, FOM snapshots are projected onto the reduced linear trial subspace, generated by means of rSVD, before being fed as input to the AE. The main advantage is that solving the POD-G ROM is more efficient than the solution of the FOM, thus we can use it to generate all the snapshots required by the POD-G DL-ROM neural network to be able to generalize to new, unseen scenarios by remarkably decreasing the time required for the generation of the training/testing datasets. On the other hand, the quality of the reduced solution is influenced by the number of retained POD modes, thus the preparation of the POD-G ROM is crucial.

Summarizing the procedure, the POD-G DL-ROM exploits the POD-G ROM in the training stage to: (1) finely sample the parameter space to guarantee the proper generalization capabilities of the DL-ROM neural network; (2) perform a dimensionality reduction of the FOM with consequent advantages in memory management and efficiency during the training stage of the DL-ROM; (3) compute the POD basis matrix employed to recover \mathbf{u}_h . The DL-ROM is still used to approximate the FOM solution during the online, testing stage, thus its efficiency and generalization capabilities are fully exploited. As result, we get a real-time simulation of the device dynamics, whose performance is detailed in Section 4.

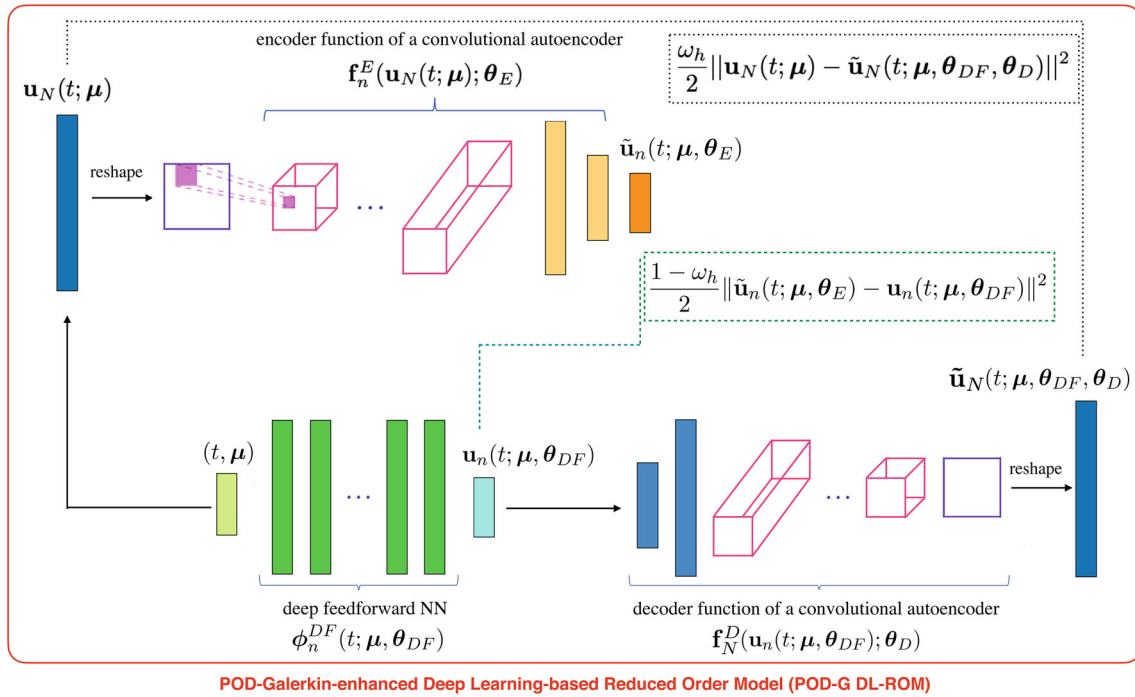


FIGURE 2 POD-G DL-ROM architecture. The approximated intrinsic coordinates $\mathbf{u}_N(t; \boldsymbol{\mu})$ are provided as input to the encoder function which outputs $\tilde{\mathbf{u}}_n(t; \boldsymbol{\mu})$. The same parameter instance associated to the POD-G ROM reduced solution, that is, $(t; \boldsymbol{\mu})$, enters the DFNN which provides as output $\mathbf{u}_n(t; \boldsymbol{\mu})$, and the error between the low-dimensional vectors is accumulated. The minimal coordinates $\mathbf{u}_n(t; \boldsymbol{\mu})$ are given as input to the decoder function returning the approximated reduced solution $\tilde{\mathbf{u}}_N(t; \boldsymbol{\mu})$. Then, the reconstruction error is computed. The approximation of the FOM solution is then recovered as $\tilde{\mathbf{u}}_h = \mathbf{V}\tilde{\mathbf{u}}_N(t; \boldsymbol{\mu})$.

3.3 | Frequency versus phase control

As recalled in Section 1, we aim at simulating the FRF of MEMS. However, the intrinsic nonlinearities imply that the system response may not be uniquely defined for a given external frequency ω and load β as a consequence of bifurcations, for example, saddle-node bifurcations. The steady-state response of the system will converge to a certain branch depending on the initial conditions. This is the case, for instance, of the simple clamped-clamped beam analyzed in Section 4.2 which is representative of a large class of resonating MEMS and it is characterized by a FRF like the one in Figure 5. For a given load multiplier β and angular frequency ω , the maximum midspan displacement over the period $T = 1/\omega$ of the beam can have one or three amplitude values, due to the so-called hardening effect. This portrait of the dynamics may be further complicated in presence of internal resonances where the number of overlapping branches and alternative solutions increases.

In this contribution, we will focus on systems without internal resonances and forced with a single frequency excitation close to an eigenfrequency. In these conditions, the dynamics is similar to the one of a simple Duffing resonator:

$$\ddot{u} + \omega_0^2 u + \varepsilon(2\xi\omega_0\dot{u} + k_3 u^3) = \varepsilon\beta \cos(\omega t), \quad (14)$$

where ω_0 is the eigenfrequency, ξ is damping coefficient, k_3 cubic nonlinearity coefficient, β is the load multiplier and $\varepsilon \ll 1$ is a *bookkeeping* parameter that sorts the order of magnitude of the terms.⁶² The multiple scales solution of Equation (14) reads:

$$u(t) = a(\beta) \cos(\omega t - \phi) + \varepsilon a^3(\beta) \frac{k_3}{32\omega_0^2} \cos(3(\omega t - \phi)) + O(\varepsilon^3), \quad (15)$$

which is plotted in several forms in Figure 3. In particular, we remark that the amplitude $a(\beta)$ can be expressed as a single-valued function of the phase $\phi \in [0 : \pi]$. It is worth recalling that this property of Duffing-like systems

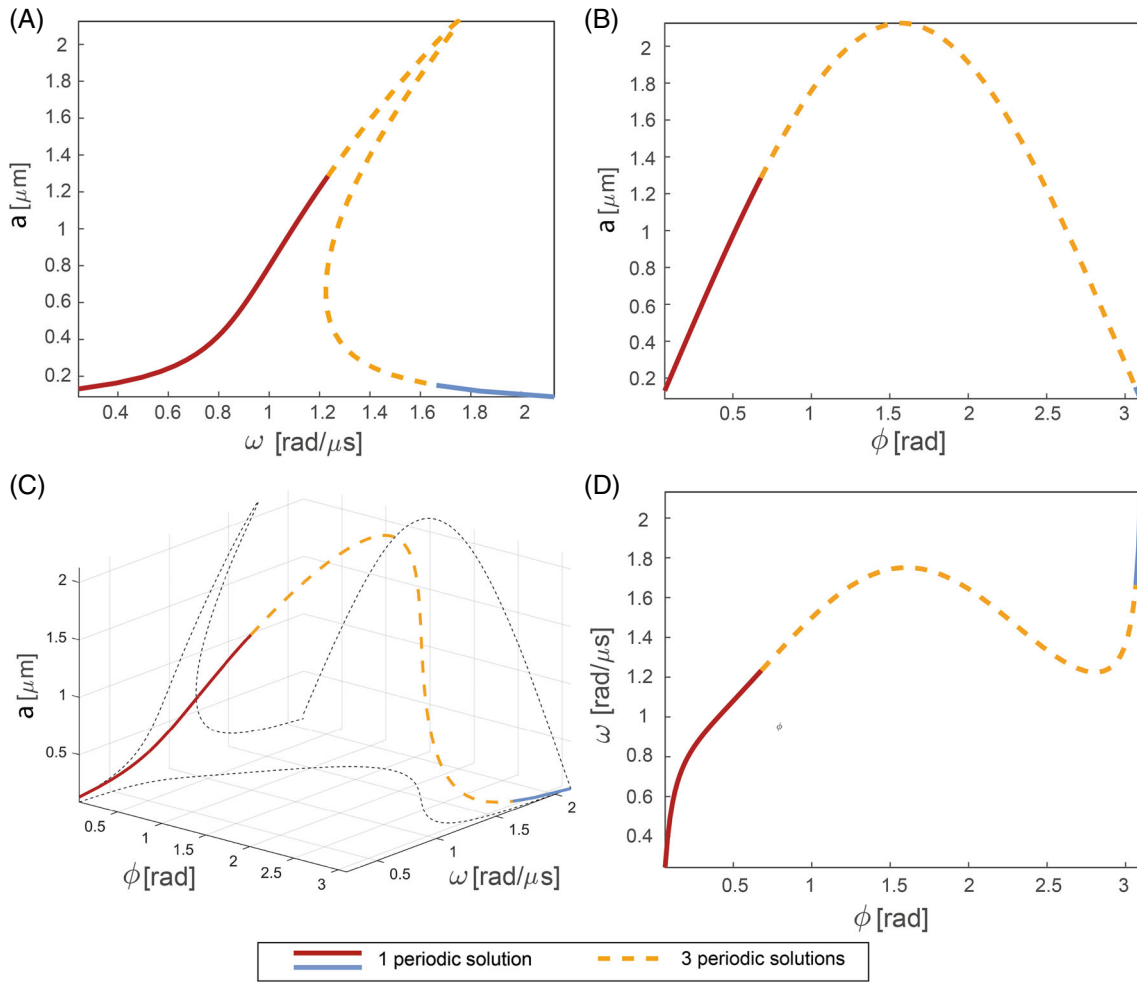


FIGURE 3 Response of the Duffing oscillator Equation (14) for $\omega_0=1$ (rad/μs), $\epsilon=1$, $\beta = 0.2$, $k_3=0.5$ and $\xi=0.05$. (A) Typical FRF, where a is plotted versus the forcing angular frequency ω . Three regions can be identified: two regions (red and blue lines, respectively) characterized by a single periodic solution for a given ω , and a region (orange line) where three solutions exist. This plot can be viewed as the projection on the $\omega - a$ plane of the response in the 3D space $\phi - \omega - a$ (C). (B) Projection of the system response of (C) on the $\phi - a$ plane, where uniqueness is preserved for any phase value. (D) Projection of the system response of (C) on the $\phi - \omega$ plane.

is exploited experimentally in phase-controlled closed-loop experiments that allow tracking both stable and unstable branches.^{63,64}

Since the POD modes in \mathbf{V} , Equation (5), are energetically ordered and we know that the first POM is generally associated with the excited eigenmode, the corresponding dof in the POD-G ROM can be used to define the phase ϕ . Moreover, the periodic solution is expressed as the sum of several harmonic components, each with a specific phase value, see for example, Equation (15). We hence define the phase referring to the lowest harmonic component which is typically the largest and also completely characterizes linear problems.

Consequently, our problem will no longer be expressed directly as a function of the parameters $\mu = [\beta, \omega]$, but rather of $\mu = [\beta, \phi]$. However, since we aim at plotting FRFs, the relationship between ϕ and ω must be suitably approximated for each value of β . Coherently with the approach taken in this investigation, we use a DFNN to model the relationship $(\phi, \beta) \mapsto \omega(\phi, \beta)$. The training data of the DFNN are represented by all the ω values at which the POD-G ROM solutions are sampled to feed the DL-ROM. This way to proceed allows solving the lack of uniqueness of the response, but nevertheless the $\omega(\phi, \beta)$ function has two asymptotes for $\phi = 0, \pi$, thus the regions close to these points are difficult to model with a DFNN. The best results have been obtained applying the following strategy. Instead of using directly ϕ in the DFNN, we formulate the input of the neural network in terms of $\sin \phi$ and $\cos \phi$. Then, the goal is to learn the function

$$f_{\phi-\omega}^{DF} : \mathbb{R}^3 \rightarrow \mathbb{R} \quad \text{such that} \quad (\sin \phi, \cos \phi, \beta) \mapsto f_{\phi-\omega}^{DF} \approx \omega(\phi, \beta). \quad (16)$$

This simple feature engineering, based on the qualitative assumptions on the problem considered, allows mitigating the effects close to the boundaries, as shown in the following section.

4 | NUMERICAL RESULTS

We discuss the application of the proposed POD-G DL-ROM technique to MEMS devices: a clamped-clamped beam resonator and a micromirror. The FOM is solved through the HB method implemented, together with a continuation scheme, in a custom Fortran library⁴⁹ for large scale problems. The generation of POD-G ROMs for MEMS has been extensively discussed by Gobat et al.²⁷ Here the ROM is solved with the Matlab package MANLAB⁵⁹ that exploits the HB method and asymptotic expansion with continuation of periodic orbits.

4.1 | Settings of neural networks

Data normalization and standardization enhance the training phase of the network by rescaling all the dataset values to a common frame. For this reason, the inputs and the output of DL-ROM are rescaled in the range $[0, 1]$ by applying an affine transformation. In particular, provided the parameter matrix $\mathbf{L} = [\mathbf{L}^{\text{train}}, \mathbf{L}^{\text{val}}] \in \mathbb{R}^{(n_\mu+1) \times N_s}$, we first rescale the time variable to the range $[0, 2\pi]$ as $\tau = t\omega$, where τ is the dimensionless time variable, and then we define

$$L_{\max}^i = \max_{j=1, \dots, N_s} L_{ij}^{\text{train}}, \quad L_{\min}^i = \min_{j=1, \dots, N_s} L_{ij}^{\text{train}}, \quad i = 1, \dots, n_\mu + 1,$$

so that data are normalized by applying the following transformation

$$L_{ij}^{\text{train}} \mapsto \frac{L_{ij}^{\text{train}} - L_{\min}^i}{L_{\max}^i - L_{\min}^i}, \quad i = 1, \dots, n_\mu + 1, j = 1, \dots, N_s. \quad (17)$$

Each feature of the training parameter matrix is rescaled according to its maximum and minimum values. Regarding instead the training snapshot matrix $\mathbf{S}_u = [\mathbf{S}_u^{\text{train}}, \mathbf{S}_u^{\text{val}}] \in \mathbb{R}^{N_h \times N_s}$, we define

$$S_{u,\max} = \max_{i=1, \dots, N_h} \max_{j=1, \dots, N_s} S_{u,ij}^{\text{train}}, \quad S_{u,\min} = \min_{i=1, \dots, N_h} \min_{j=1, \dots, N_s} S_{u,ij}^{\text{train}},$$

and apply transformation Equation (17) by replacing L_{\max}^i, L_{\min}^i with $S_{u,\max}, S_{u,\min}$, respectively, that is we use the same maximum and minimum values for all the features of the snapshot matrix. Transformation Equation (17) is applied also to the validation and testing sets, but considering as maximum and minimum the values computed over the training set. In order to rescale the reconstructed solution to the original values, we apply the inverse transformation of Equation (17). The configuration of the DL-ROM neural network used in the following test cases is the one given below. We choose a 12-layer DFNN equipped with 50 neurons per hidden layer and n neurons in the output layer, where n represents the dimension of the (nonlinear) reduced trial manifold. The architectures of the encoder and decoder functions are instead reported in Tables 1 and 2. No activation function is applied at the last convolutional layer of the decoder neural network, as usually done in AEs. To solve the optimization problem Equations (12) and (13), we use the ADAM algorithm,⁶⁵ which is a stochastic gradient descent method computing an adaptive approximation of the first and second momentum of the gradients of the loss function. In particular, it computes exponentially weighted moving averages of the gradients and of the squared gradients. We set the starting learning rate to $\eta = 10^{-4}$, the batch size to 40, and perform cross-validation in order to tune the hyperparameters of the DL-ROM, by splitting the data in training and validation sets with a proportion 8:2. Moreover, we implement an early stopping regularization technique to reduce overfitting,⁶⁶ stopping the training if the loss does not decrease over 500 epochs. The maximum number of epochs is set to 10,000. As nonlinear activation function, we employ the ELU function.⁶⁷ The parameters, weights and biases, are initialized through the He uniform initialization.⁶⁸ The interested reader can refer to Reference 56 for a detailed version of the algorithms used for the training/testing phases. The latter have been carried out on a Tesla V100 32GB GPU by means of the Tensorflow DL framework.⁶⁹

TABLE 1 Attributes of convolutional and dense layers in the encoder \mathbf{f}_n^E .

Layer	Input dimension	Output dimension	Kernel size	#of filters	Stride	Padding
1	[N, N, 1]	[N, N, 8]	[5, 5]	8	1	SAME
2	[N, N, 8]	[N/2, N/2, 16]	[5, 5]	16	2	SAME
3	[N/2, N/2, 16]	[N/4, N/4, 32]	[5, 5]	32	2	SAME
4	[N/4, N/4, 32]	[N/8, N/8, 64]	[5, 5]	64	2	SAME
5	N	64				
6	64	n				

TABLE 2 Attributes of dense and transposed convolutional layers in the decoder \mathbf{f}_N^D .

Layer	Input dimension	Output dimension	Kernel size	#of filters	Stride	Padding
1	n	256				
2	256	N_h				
3	[N/8, N/8, 64]	[N/4, N/4, 64]	[5, 5]	64	2	SAME
4	[N/4, N/4, 64]	[N/2, N/2, 32]	[5, 5]	32	2	SAME
5	[N/2, N/2, 32]	[N, N, 16]	[5, 5]	16	2	SAME
6	[N, N, 16]	[N, N, 1]	[5, 5]	1	1	SAME

The reduction process requires training the $f_{\phi-\omega}^{DF}$ DFNN as well. The architecture of the NN is the same for both resonator and micromirror. The DFNN receives as input cosine and sine values of a phase ϕ and a load multiplier β and gives as result a frequency ω . Since the solutions are computed with HB method the calculation of the phase response ϕ on the first harmonic component is straightforward since it corresponds to $\tan(a_1/b_1)$ with a_1 amplitude of the first harmonic sine and b_1 amplitude of the first harmonic cosine. In the next sections, we will plot the results of the $f_{\phi-\omega}^{DF}$ DFNN using the phase ϕ instead of the cosine and sine for sake of clarity. During the online stage this feature extraction preprocess is embedded in the neural network input, thus only a value for the phase ϕ and a forcing level β are provided. The function $f_{\phi-\omega}^{DF} : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a 10-layer DFNN equipped with 64 neurons per hidden layer and 1 neuron in the output layer. We employ the tanh activation function, Adam optimization algorithm, mean squared error metric, and batch size equal to 20. The starting learning rate to $\eta = 10^{-3}$ and perform cross-validation in order to tune the hyperparameters NN, by splitting the data in training and validation sets with a proportion 1:1. The parameters, weights and biases, are initialized through the Glorot normal initialization.⁷⁰ The early stopping criterion is used to determine the best fit with a patience coefficient of 50 epochs over a total number of epochs equal to 2000. The implementation was carried out in the Tensorflow DL framework.⁶⁹

4.2 | Doubly clamped beam resonator

Let us consider a doubly clamped beam of length $L = 1000 \mu\text{m}$ with a rectangular cross-section of dimensions $10 \mu\text{m} \times 24 \mu\text{m}$, as depicted in Figure 4A. Even though this example is proposed primarily to discuss the performance of the proposed technique in the simplest possible setting, nevertheless it also emulates realistic MEMS resonators like those analyzed in Reference 63 and depicted in Figure 4B,C. The beam is made of isotropic polysilicon,⁷¹ with density $\rho = 2330 \text{ kg/m}^3$, Young modulus $E = 167 \text{ GPa}$ and Poisson coefficient $\nu = 0.22$.

Dirichlet boundary conditions ($\mathbf{u}^D = \mathbf{0}$) are applied on the two opposite sides of the beam. Neumann boundary conditions ($\mathbf{f}^D = \mathbf{0}$) are enforced elsewhere. The quality factor is assumed equal to $Q = 50$. The device is excited, for simplicity, by a body load $\mathbf{F} = \mathbf{M}\phi_1\beta \cos(\omega t)$ proportional to the first eigenmode ϕ_1 , with \mathbf{M} mass matrix and β load multiplier. The discussion aims at demonstrating the accuracy of the POD-G DL-ROM technique by proposing comparisons with the HB-FOM. Hence, in order to keep the FOM computational time at a reasonable level, a rather coarse mesh with 2607 nodes has been employed. The first five eigenfrequencies are reported in Table 3.

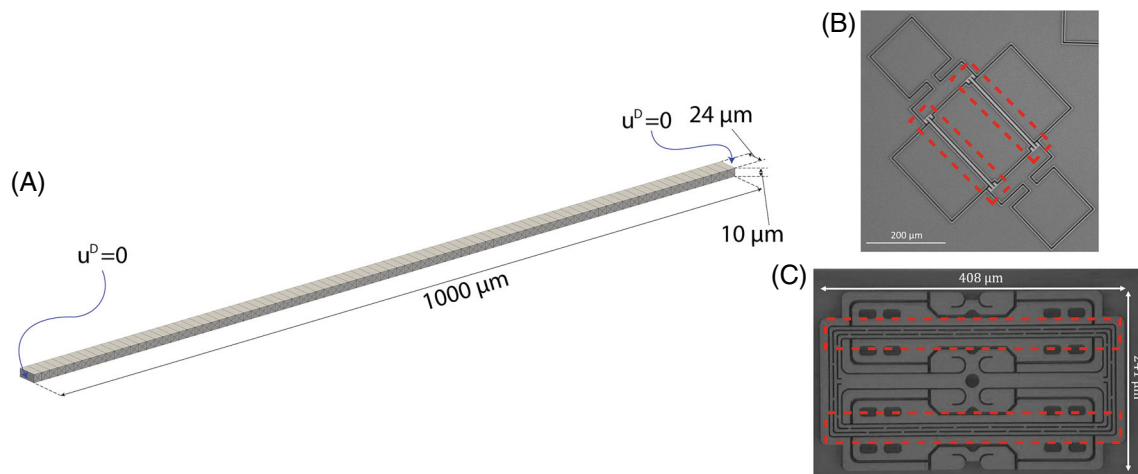


FIGURE 4 Clamped-clamped beam. (A) Geometry and mesh of the doubly clamped beam, 3D view. This academic example can be considered as a simplified version of real MEMS resonators like the ones in (B, C) analyzed by Zega et al..⁶³ The red boxes frame the beam resonators in the picture.

TABLE 3 Doubly clamped beam: First 5 eigenfrequencies of the FEM model.

Eigenmode	1	2	3	4	5
Frequency (kHz)	87.141	208.45	240.03	470.10	572.18

The POD-G ROM basis vectors are computed using 500 high-fidelity snapshots generated from one period of 10 frequency samples, picked along the FRF corresponding to $\beta = 0.5 \mu\text{N}$. The sampling values are represented by the violet circle markers in Figure 5A. The POD-G ROM is built retaining in the linear trial space only the first 4 most energetic POD modes, depicted in Figure 5B–E, respectively. This number of POD modes represents the minimum required to achieve a good accuracy while keeping the model relatively small.

A comprehensive analysis of the POD-G applied to this benchmark has been proposed by authors in Reference 27 where in particular a discussion on the convergence with respect to the number of samples and of POD modes retained is presented. In this contribution we will rather focus on the subsequent phases of the procedure. However, for the sake of completeness, in Figures 6, 8, and 9 also the POD-G results, together with those of the DL-ROM procedure, are compared against the FOM in order to provide a critical assessment of the POD-G procedure in our applications.

Next, the POD-G ROM is used to generate the training data for the DL-ROM.

The training dataset is represented in Figure 5F where the red markers on the FRFs denote the combinations of forcing and phase employed, spanning the space (β, ϕ) in $\{0.25, 0.375, 0.5, 0.625, 0.75\} \mu\text{N} \times [0.1714 : 3.1344]$ rad with a total of 309 combinations. For each phase-frequency combination 50 snapshots have been taken along each simulated period, yielding a total of 15,450 snapshots. A convergence analysis with respect to the number of training points in this phase will be addressed later in Figure 10.

To better understand the efficacy of the procedure we compare, in Figure 6, the solutions obtained with the FOM, the POD-G ROM and the POD-G DL-ROM approaches in terms of the plots $\omega(\phi)$, $u_{\text{mid}}(\phi)$ (with u_{mid} maximum midspan displacement) and $u_{\text{mid}}(\omega)$ (the FRFs) for different β values.

The plots of the frequency-phase functions are reported in Figure 6A while enlarged views of specific regions are proposed in Figure 6B,C. The FOM, POD-G ROM, and POD-G DL-ROM solutions are nearly superimposed along the whole curves, some differences being observed where the phase approaches the vertical asymptotes. Only minor differences can be observed at higher amplitudes which can be ascribed to the limited number (4) of POD modes retained in the ROM. The POD-G DL-ROM induces a slight increment in the error with respect to the FOM. Nevertheless, its accuracy is still excellent. The phase-amplitude functions are presented in Figure 6D, while enlarged views are reported in Figure 6E,F. Also in this case FOM, POD-G ROM, and POD-G DL-ROM solutions are nearly superimposed, even at the boundaries of the phase parameter space. The FRF given by the combination of the previous functions is shown in Figure 6G, with

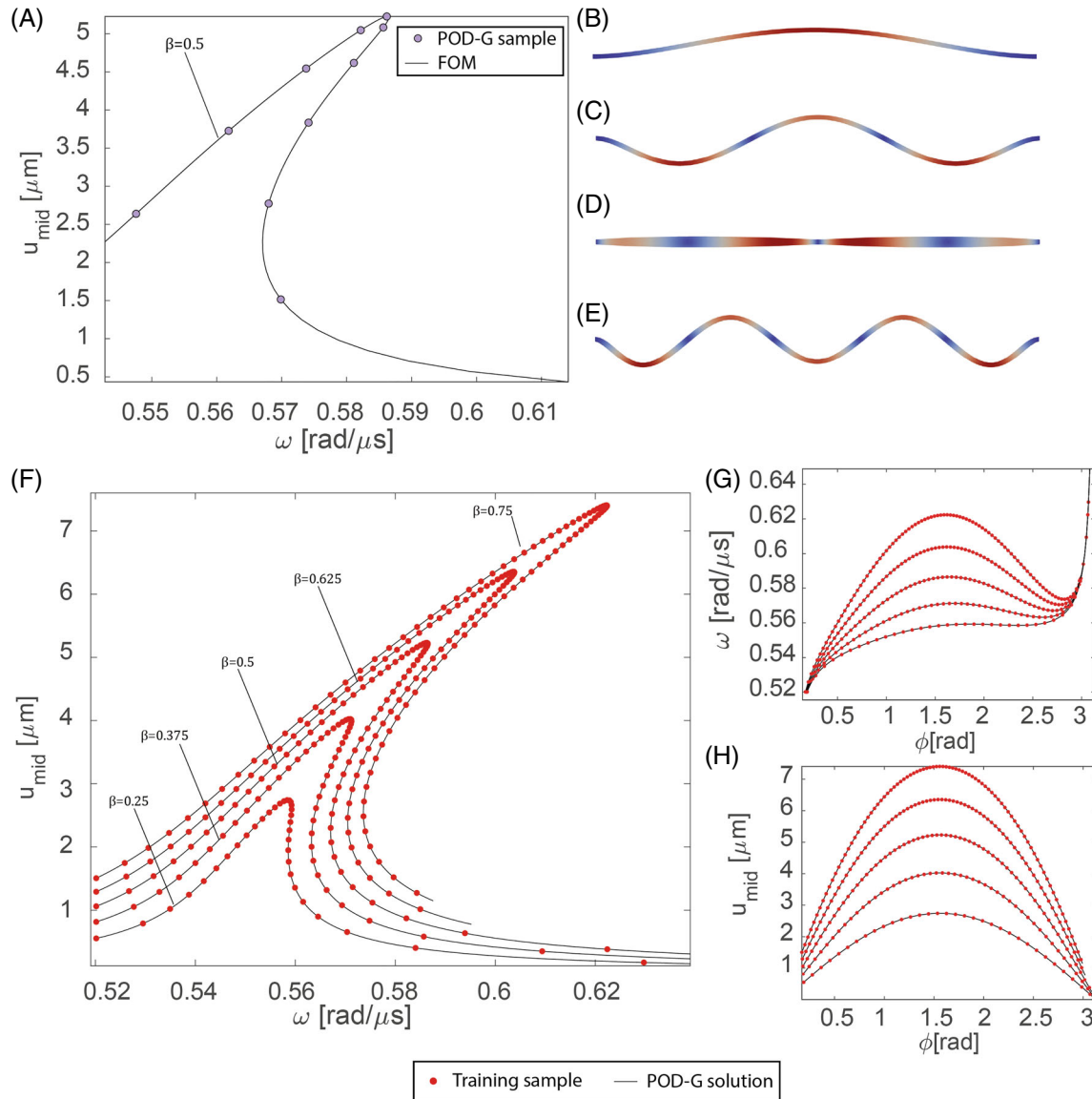


FIGURE 5 Clamped-clamped beam. (A) FOM sampling points ($\beta = 0.5$) used to generate the snapshots for the POD-G ROM. (B–E) first four most energetic POD modes obtained from SVD. (F) FRFs $u_{\text{mid}}(\omega)$ computed with the POD-G ROM for different load multipliers β ; red markers identify the training dataset used for the DL-ROM (some points have been clipped at higher frequencies). (G, H) Plots of the $\omega(\phi)$ and $u_{\text{mid}}(\phi)$ functions corresponding to the FRFs of (F), with the same β values.

enlarged views being presented in Figure 6H,I. We notice that the resulting FRFs are almost identical and deviate only very close to the limits of the $\omega(\phi)$ approximations (e.g., close to $\omega = 0.59$ rad/μs on the lower branch of the FRF).

All the periodic solutions computed with the POD-G DL-ROM are represented as an envelope of all the FRFs for different values of the forcing in Figure 7. Such a plot, that would be demanding even with a POD-G ROM, can be completed with the POD-G DL-ROM in few seconds and opens the way to the use of this technique in the real time simulation and optimization of complex systems.

A further assessment of the POD-G DL-ROM performance concerns time histories along one period. In Figure 8A we plot the periodic responses of the midspan displacement corresponding to the parameter instances highlighted with the black circle markers in Figure 6, while Figure 8B,C illustrate the error of the POD-G ROM and the POD-G DL-ROM with respect to the FOM. It is worth stressing that the procedure reconstructs, for each parameter query and along the whole period, an approximation of the FOM solution everywhere on the structure and not only at selected points. To quantify the error globally introduced in the spatial domain, we first define the instantaneous relative measure for a given parameter instance:

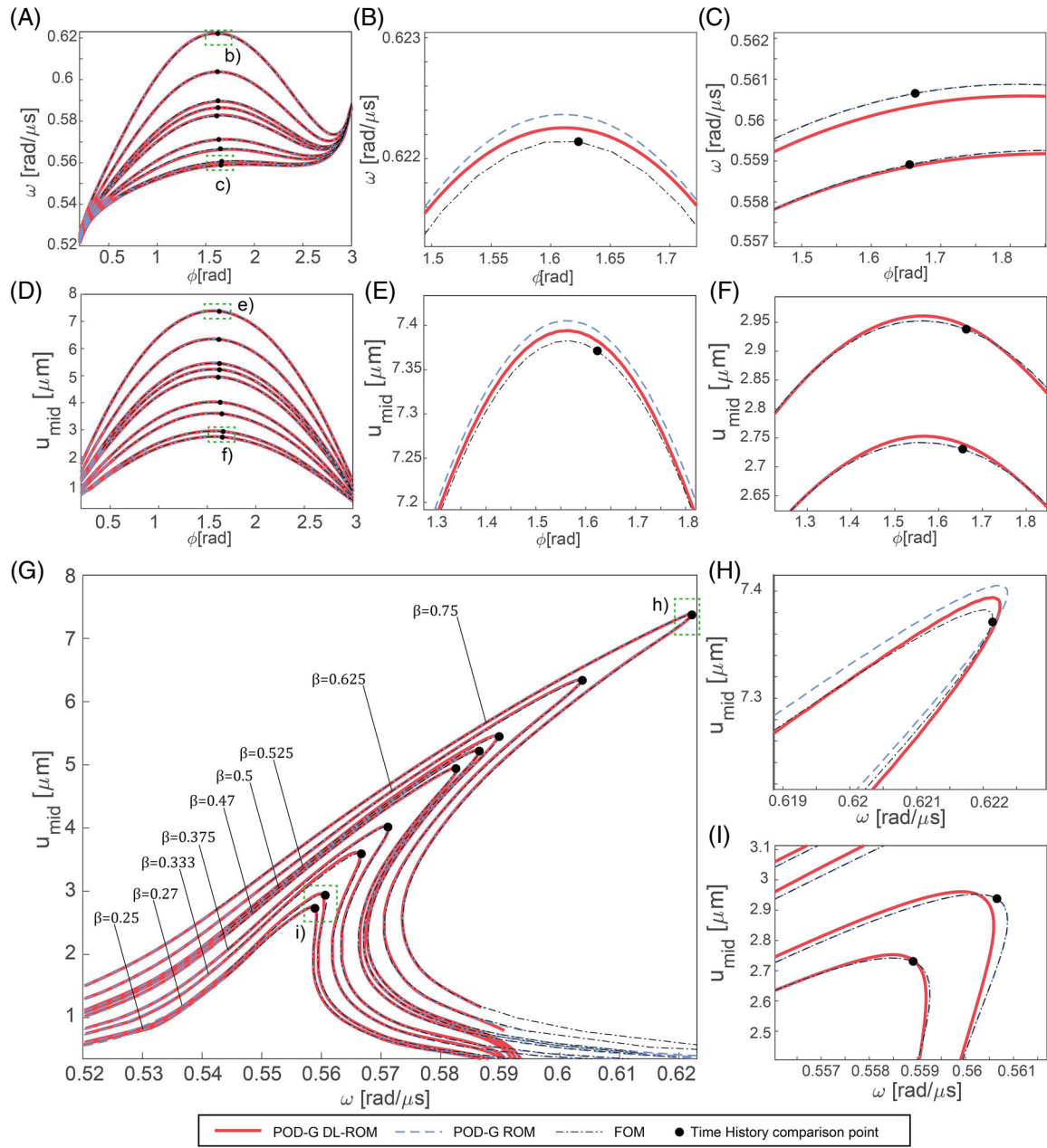


FIGURE 6 Clamped-clamped beam. Results of the POD-G DL-ROM (red lines) compared with the FOM (black dash-dotted lines) and with the POD-G ROM (dashed blue lines). (G) Final $u_{\text{mid}}(\omega)$ FRF curves for different β values. (H, I) Enlarged views of the green boxes of (G). The black dots denote points that will be further investigated in Figure 8. (A) $\omega(\phi)$ curves for the same β values of (G). (B, C) Enlarged views of the green boxes of (A). (D) $u_{\text{mid}}(\phi)$ curves for the same β values of (G). (E, F) Enlarged views of the green boxes of (D).

$$\epsilon_{FOM}(\mu, t) = \frac{\|\mathbf{u}_h(\mu, t) - \tilde{\mathbf{u}}_h(\mu, t)\|}{\sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} \|\mathbf{u}_h(\mu, t_k)\|^2}}, \quad (18)$$

where N_t denotes the number of time instances considered along one period, $\mathbf{u}_h(t)$ is the high-fidelity solution vector at time t , $\tilde{\mathbf{u}}_h(t)$ is the POD-G DL-ROM reconstruction of the FOM solution and $\|\cdot\|$ denotes the L2-norm at a given time t . The denominator of Equation (18) provides a reference value for the normalization given by the average of the L2-norm of the solution along the whole period. The relative errors, always close to or lower than $\epsilon_{FOM} = 10^{-3}$, are reported in Figure 9.

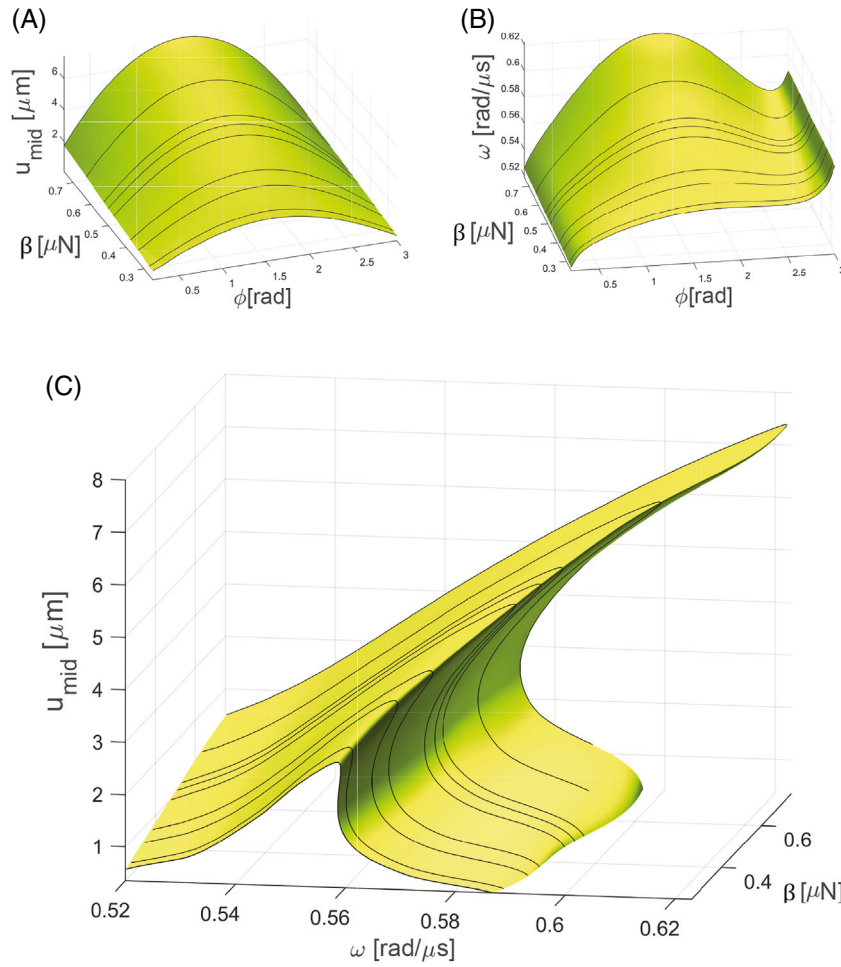


FIGURE 7 Clamped-clamped beam. Solution envelopes generated with the POD-G DL-ROM. (A) Envelop surface for $u_{\text{mid}}(\phi, \beta)$. (B) Envelop surface for $\omega(\phi, \beta)$. (C) Envelop of the FRFs $u_{\text{mid}}(\omega, \beta)$. The black lines are the POD-G DL-ROM solutions shown in Figure 6 and are here reported to better highlight the surface curvature.

Next, we compute an average relative error $\bar{\epsilon}_{FOM}$, with respect to the FOM, over all the N_t time instants of a the period for all the N_{test} parameter instances tested:

$$\bar{\epsilon}_{FOM} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} \epsilon_{FOM}^2(\mu_i, t_k)}. \quad (19)$$

The results are reported in Figure 10 where, for a variable number of training snapshots, two error measures are plotted: the loss function (13) and $\bar{\epsilon}_{FOM}$. The training set described above corresponds to the largest number of snapshots considered, and it has been progressively rarefied by halving each time the number of phase instances for each β instance. In particular, by analyzing the convergence properties of the POD-G DL-ROM technique, is evident that, by providing a sufficient amount of training data to the neural network, the level of accuracy achieved by the POD-DL-ROM is remarkable over the entire testing set (hence over the whole parameter space \mathcal{P}). Moreover, we point out that, by considering parameter instances within the parameter space and sufficiently far from the boundaries, we expect a further improvement in the accuracy of the POD-G DL-ROM approximation with respect to the FOM solution. Indeed, as highlighted in Figure 9 referring to the peaks of the FRF for the largest number of snapshots, typical relative errors are almost one order of magnitude smaller than the average reported in Figure 10.

Next, we compare the computational time required to solve the FOM with the POD-G ROM and the POD-G DL-ROM testing times, that is, the time needed to compute the ROM solution for a given number of parameter instances. The

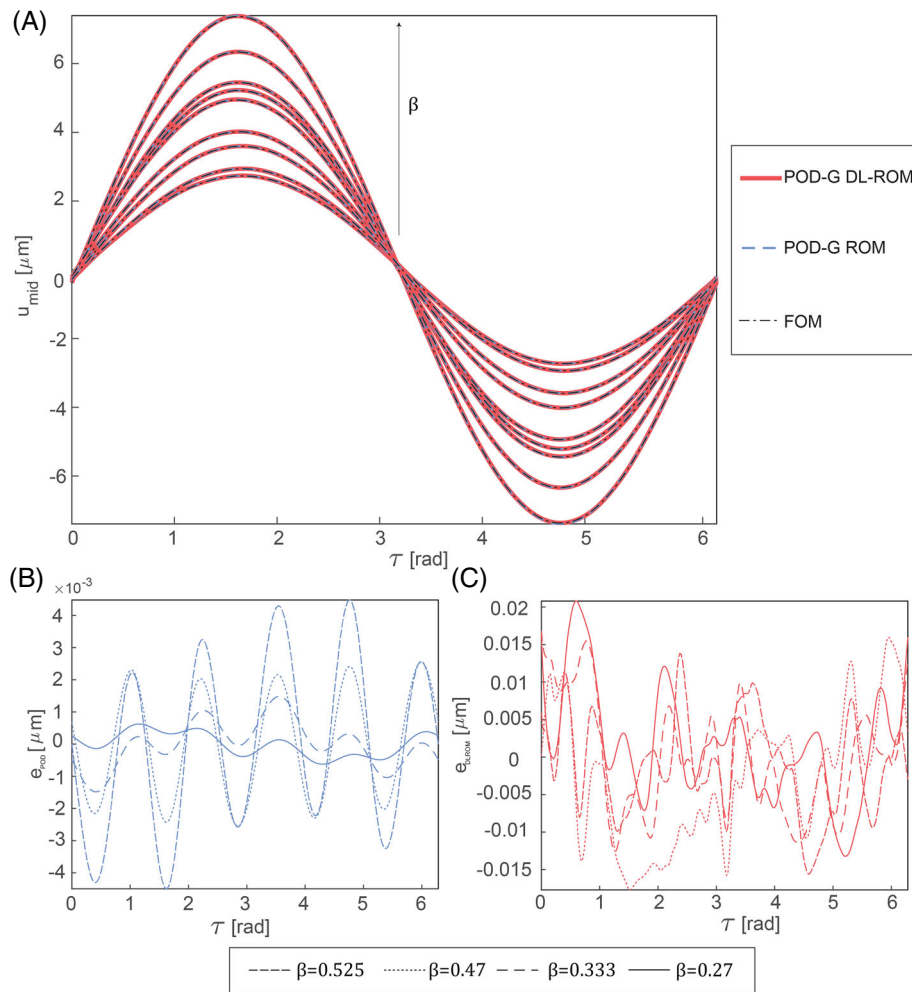


FIGURE 8 Clamped-clamped beam. Comparison between the periodic responses of the FOM (black line), the POD-G ROM (blue line), and the POD-G DL-ROM (red line) solutions. The sampling points are the ones detailed in Figure 6. (A) Depicts the system response along each period. (B, C) Represent the error of the POD-G ROM and the POD-G DL-ROM with respect to the FOM, computed as the difference between the midspan displacement instantaneous values.

results are summarized in Table 4 and the computational cost trend is depicted in Figure 11. We stress that the T_{POD} and T_{DL} data do not include the training phase, which will be performed off-line, beforehand.

We notice that the T_{FOM} explodes when increasing the number of instances while the POD-G ROM (T_{POD}) can drastically reduce the computing time making it a suitable approach when there is no need of truly real-time applications. As extensively discussed in Reference 27, in this case a robust ROM can be built with only 4 POD modes. Nevertheless, even in this extremely simple example, the POD-G DL-ROM testing time (T_{DL}) highlights the impressive performance of the technique, which guarantees a reduced effort, with respect to the FOM, of a factor 10^5 .

4.3 | MEMS micromirror

Micromirrors represent one of the most promising examples of the next generation of MEMS which will deeply impact our lives and enable a sustainable evolution of the Internet of Things. Among their virtually countless applications, one can cite pico-projectors for Augmented Reality (AR) lenses⁷² and 3D scanners for Light Detection and Ranging (LiDAR) in autonomous driving.

Because of the inertial and geometrical effects triggered by large rotations, micromirrors are intrinsically nonlinear and the prediction of their dynamic behavior is essential to guarantee a proper design and control of the mirror during

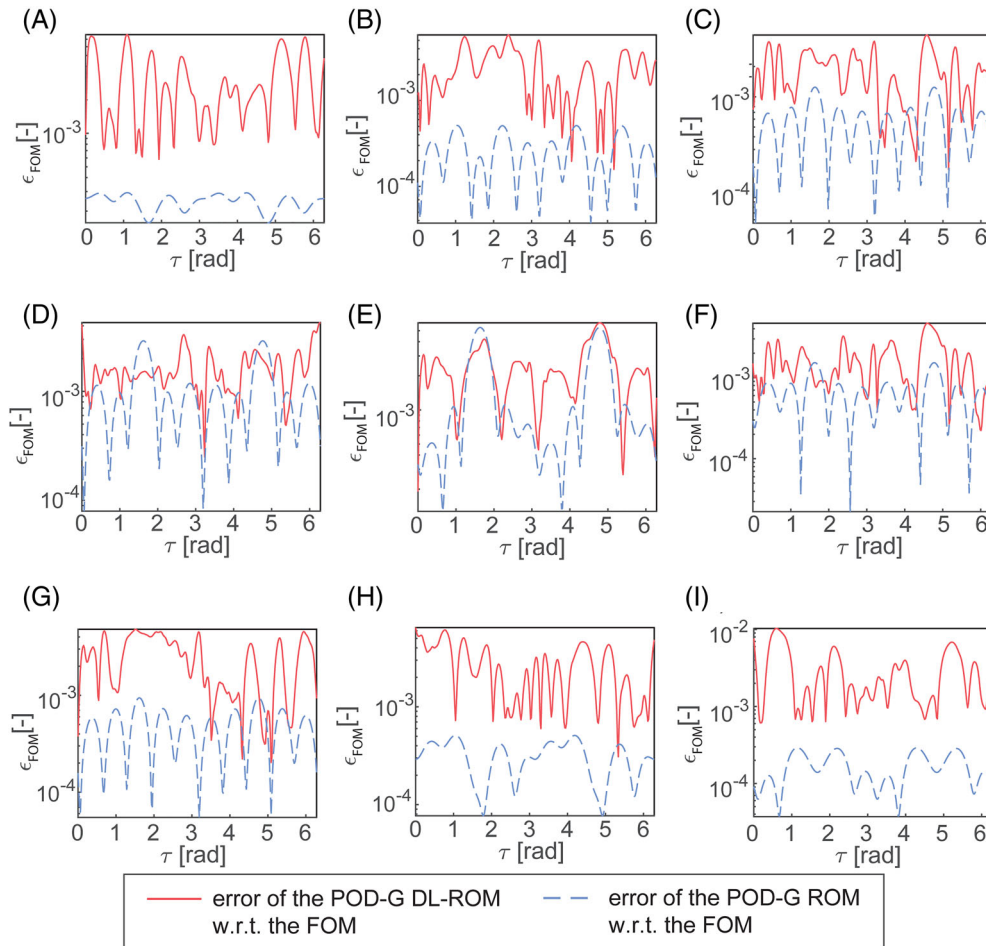


FIGURE 9 Clamped-clamped beam. Relative instantaneous error $\epsilon_{FOM}(t)$ (Equation 18) along one period with respect to the FOM solution, of the POD-G ROM (blue line) and the POD-G DL-ROM (red line) approximations. The sampling points are the ones detailed in Figure 6 and corresponds to the peaks of the FRFs. (A) $\beta = 0.25$; (B) $\beta = 0.375$; (C) $\beta = 0.5$; (D) $\beta = 0.625$; (E) $\beta = 0.75$; (F) $\beta = 0.525$; (G) $\beta = 0.47$; (H) $\beta = 0.333$; (I) $\beta = 0.27$

the online stage. Only recently their numerical simulation has been tackled by Opreni et al.⁴⁹ with a dedicated large scale FOM based on HB method. It is worth stressing that the generation of a suitable ROM is a tough challenge even for the most advanced and recent techniques, like the direct normal form approach, mainly because the torsional mode is not the lowest-frequency one (it is the third) and is not well separated from the other modes. Indeed, the quadratic formulation of the direct normal form approach implemented by Opreni et al.²⁶ fails and a high order expansion is required, as remarked by Vizzaccaro et al..⁴⁷ Similar difficulties have been experienced with the implicit condensation approach.⁷⁴

The mirror considered, fabricated by STMicroelectronics, is illustrated in Figure 12. The mirror plate is suspended to a gimbal connected with a torsional beam along the rotation axis and two suspension beams on each side. The mirror is assumed to be made of isotropic polysilicon,⁷¹ with density $\rho = 2330 \text{ Kg/m}^3$, Young modulus $E = 167 \text{ GPa}$ and Poisson coefficient $\nu = 0.22$. Since the central plate is very stiff, we adopt its angle of rotation θ as reference for the FRFs amplitude and for the time response. Thanks to symmetry, only half of the micromirror is modeled with the FEM and a total of 9732 dofs. The Dirichlet boundary conditions are imposed on the substrate (the dark green areas in Figure 12B and on the symmetry plane). On the remaining boundaries, zero traction Neumann boundary conditions are imposed. The first five eigenfrequencies obtained from a linear FEM eigenvalue analysis are listed in Table 5; the torsional mode of interest herein, the third one, has a frequency of 29271 Hz. A quality factor equal to $Q = 1000$ is considered.

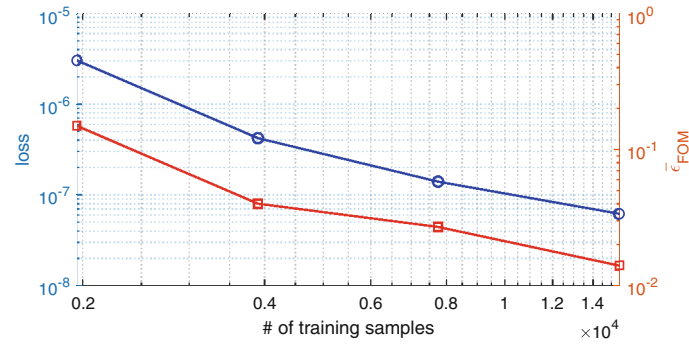


FIGURE 10 Clamped-clamped beam. Convergence analysis with respect to the number of training samples. Two errors measures on testing instances are reported. The left axis refers to the loss (blue curve) while the right axis to the average relative error (red curve). The axes are in logarithmic scales.

TABLE 4 Clamped-clamped beam. Computational cost of the FOM (T_{FOM}), POD-G ROM (T_{POD}), and POD-G DL-ROM (T_{DL}) and speedup of the reduction techniques. In the latter two cases we report only the testing time, without including the training phase.

Number of instances	T_{FOM}	T_{POD}	T_{DL}	$T_{\text{FOM}}/T_{\text{DL}}$	$T_{\text{POD}}/T_{\text{DL}}$
300	212s	0.009s	0.0015s	1.41×10^5	6.0
10^5	20.13h	3.2s	0.18s	4.0×10^5	17.7
10^6	8.4d	31.5s	2.0s	3.62×10^5	15.7
6×10^7	503d	1890s	115s	3.77×10^5	16.4

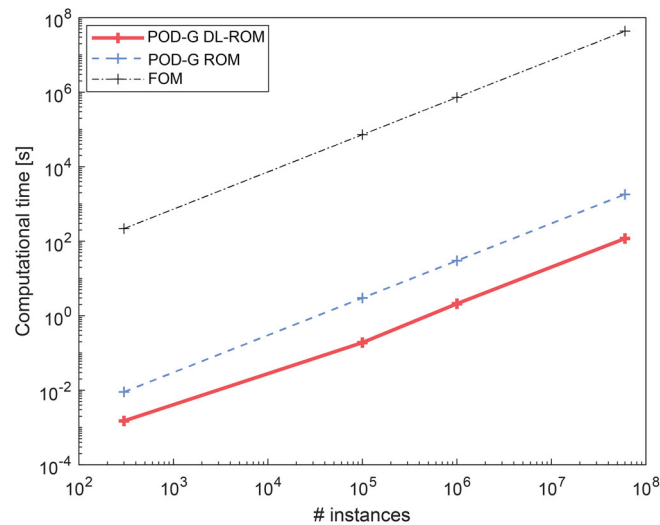


FIGURE 11 Clamped-clamped beam. Plot of the computational time as a function of the number of instances for the FOM, POD-G ROM, and POD-G DL-ROM. All the methods display a linear trend.

TABLE 5 Micromirror: First five eigenfrequencies computed with a linear FEM.

Eigenmode	1	2	3	4	5
Frequency (kHz)	11.080	18.533	29.271	41.667	68.848

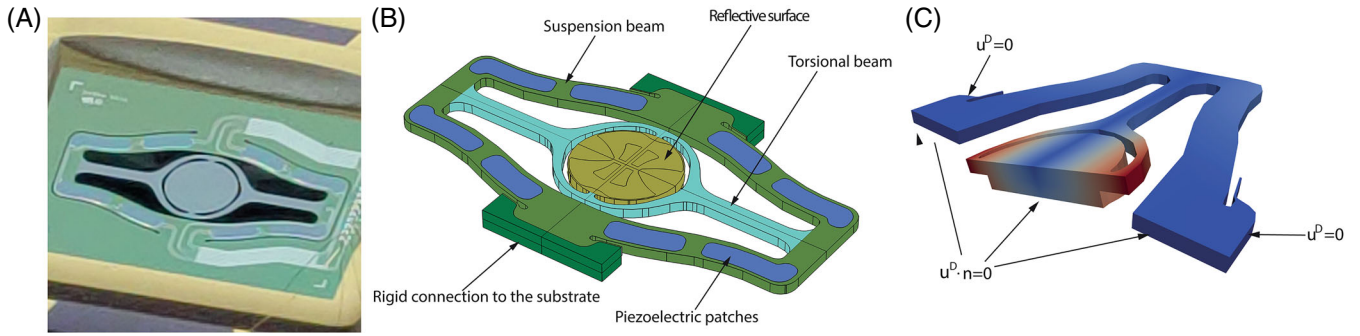


FIGURE 12 Micromirror. (A) Optical picture of the micromirror. (B) Schematic view of the layout with few details on the device components.^{49,73} (C) Third (torsional) eigenmode that is actuated during operations and the boundary conditions used in the simulation of half of the device.

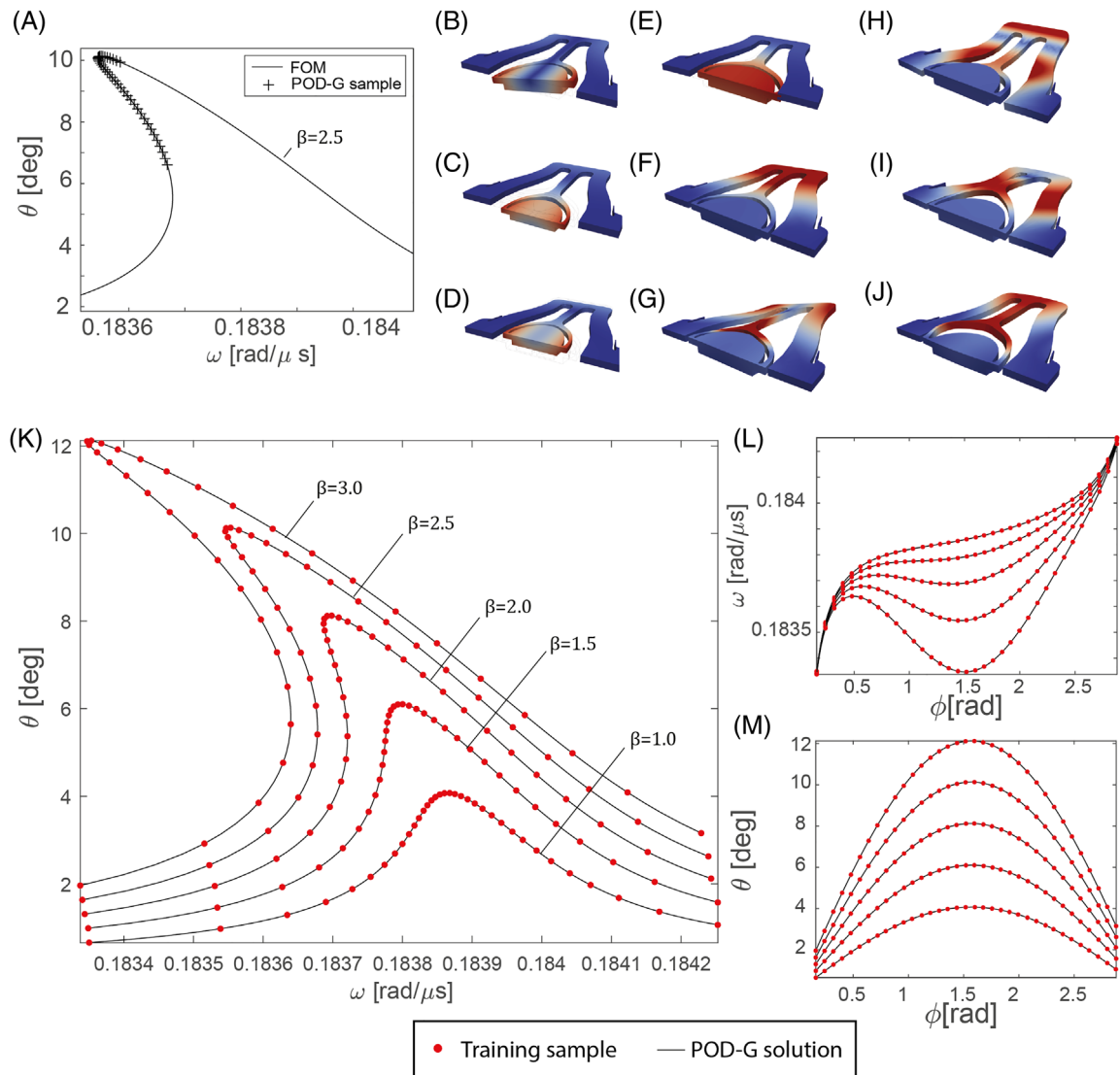


FIGURE 13 Micromirror. (A) Sampling points on the FOM FRF used to build the POD-G ROM subspace bases. (B–J) 9 most energetic POD modes used in the reduced basis. (K) FRFs $\theta(\omega)$ computed with the POD-G ROM for different β values. The training sampling points used to generate the snapshots for the DL-ROM construction are highlighted with red markers. Some points at higher frequencies are clipped. (L, M) $\omega(\phi)$ and $\theta(\phi)$ functions corresponding to the FRFs of (K), with the same β values.

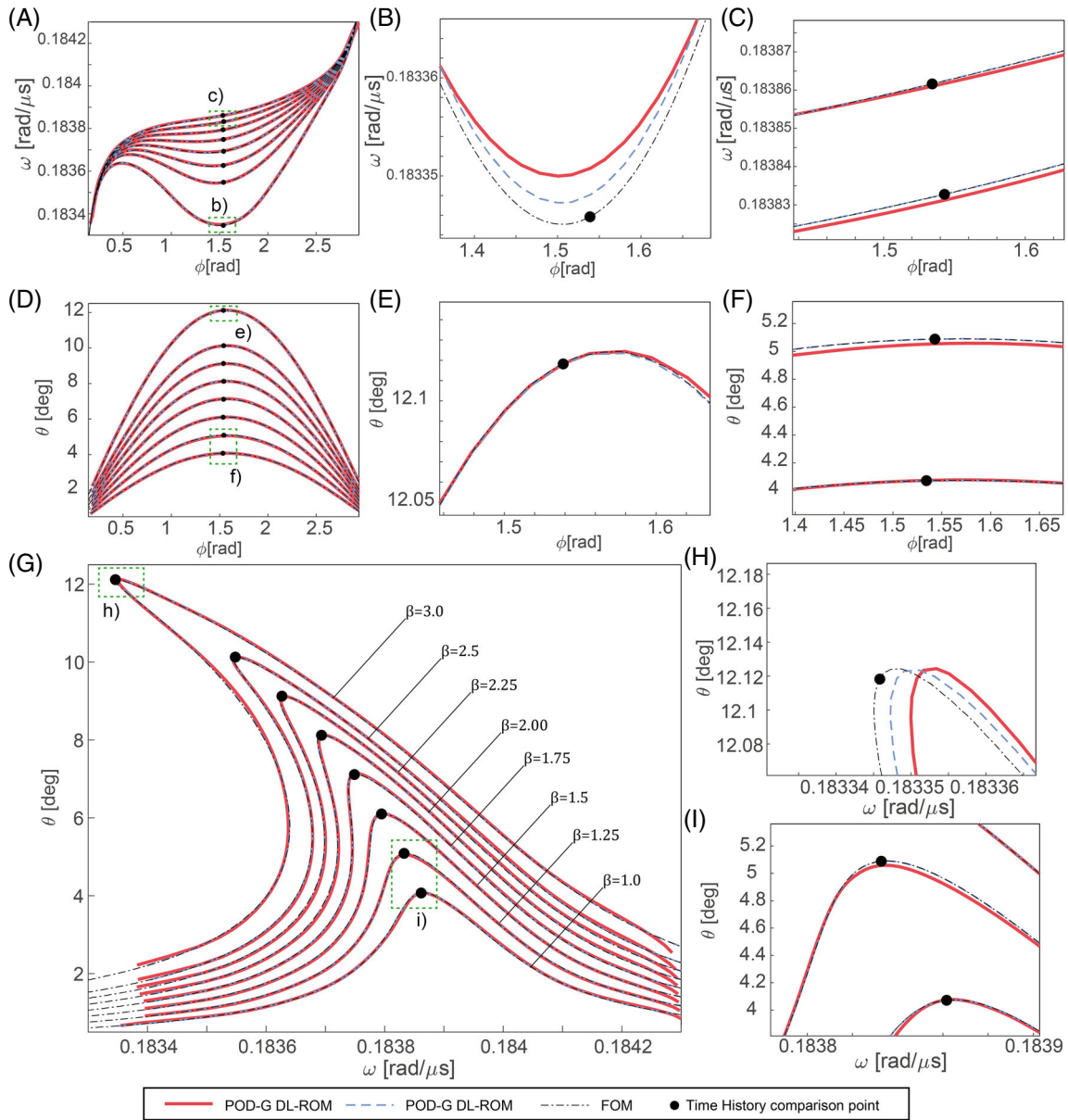


FIGURE 14 Micromirror: Results of the POD-G DL-ROM (red lines) compared with the FOM (black dash-dotted lines) and with the POD-G ROM (dashed blue lines). (A, D) Frequency $\omega(\phi)$ and angle $\theta(\phi)$ plots, respectively, for various forcing levels. (B, C) are enlarged views of $\omega(\phi)$ in (A). (E, F) are enlarged views of $\theta(\phi)$ in (B).

The micromirror is inspired by the one analyzed by Opreni et al.,⁴⁹ nevertheless we simplify the forcing mechanism by replacing the piezoelectric actuation with a body load proportional to the third torsional eigenmode $\mathbf{F} = \mathbf{M}\boldsymbol{\phi}_3\beta \cos(\omega t)$. Such a simplification allows focusing only on the application of the POD-G DL-ROM technique to the modeling of geometric and inertia nonlinearities. An HB-FOM is used to generate the high-fidelity solutions to be processed with SVD. We collect 2000 snapshots obtained for $\beta = 2.5 \mu\text{N}$ and 40 frequency values distributed as illustrated by the cross markers in Figure 13A. To achieve a good balance between subspace dimension and quality of the solution, the 9 POD modes illustrated in Figure 13B–J are retained in the reduced basis.

A critical analysis of the POD-G applied to this benchmark has been proposed by the authors in Reference 27. Anyway, in Figures 14, 16, and 17 we also report the relative error of POD-G results with respect to the FOM, as a further check.

The POD-G ROM is next used to generate the low-fidelity snapshots for the DL-ROM and the $f_{\phi-\omega}^{DF}$ DFNN.

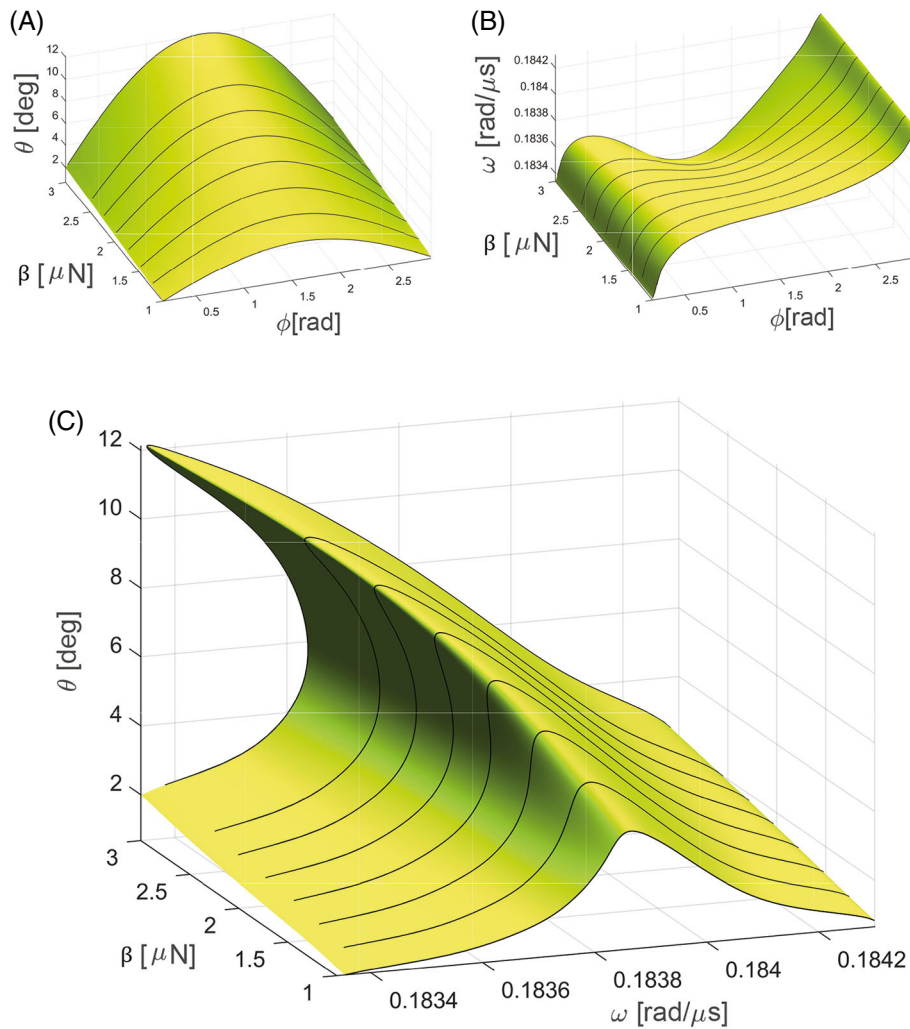


FIGURE 15 Micromirror. Envelop of the solutions reconstructed with the POD-G DL-ROM. (A) $\theta(\phi; \beta)$ envelop. (B) $\omega(\phi; \beta)$ envelop. (C) $\theta(\omega; \beta)$ envelop. The black line are the POD-G DL-ROM solutions shown in Figure 6 and are here reported to better represent the surface curvature.

The training dataset is represented in Figure 13K where the red markers on the FRFs denote the combinations of forcing and phase employed, spanning the space (β, ϕ) in $\{1.0, 1.5, 2.0, 2.5, 3.0\} \mu\text{N} \times [0.1604 : 2.9211] \text{ rad}$ with a total of 175 combinations. For each parameter instance, the period of the response is sampled with 100 snapshots.

The trained DL-ROM is finally applied to build the FRF curves for all the desired forcing levels as plotted in Figure 14. Similarly to what remarked in the previous application to the clamped-clamped beam, the POD-G DL-ROM reproduces the dynamic behavior of the system with remarkable accuracy and errors are limited to regions corresponding to the lower and upper bounds of the phase, of little engineering interest. Also in this case, the POD-G DL-ROM can be used to produce an almost continuous envelop of the system FRFs, as shown in Figure 15. The same accuracy is displayed by the time evolution of the angle, reported in Figure 16, corresponding to the parameter combinations highlighted by black dots in Figure 14. The error of the POD-G ROM and the POD-G DL-ROM with respect to the FOM is depicted in Figure 16B,C respectively.

Finally, the relative error ε_{FOM} , Equation (18) is plotted in Figure 17 for various load multipliers β , while Figure 18 collects a convergence analysis, in terms of the loss function (13) and of $\bar{\varepsilon}_{FOM}$. The training set described above corresponding to the largest number of snapshots considered, it is progressively coarsened by halving the number of phase points sampled.

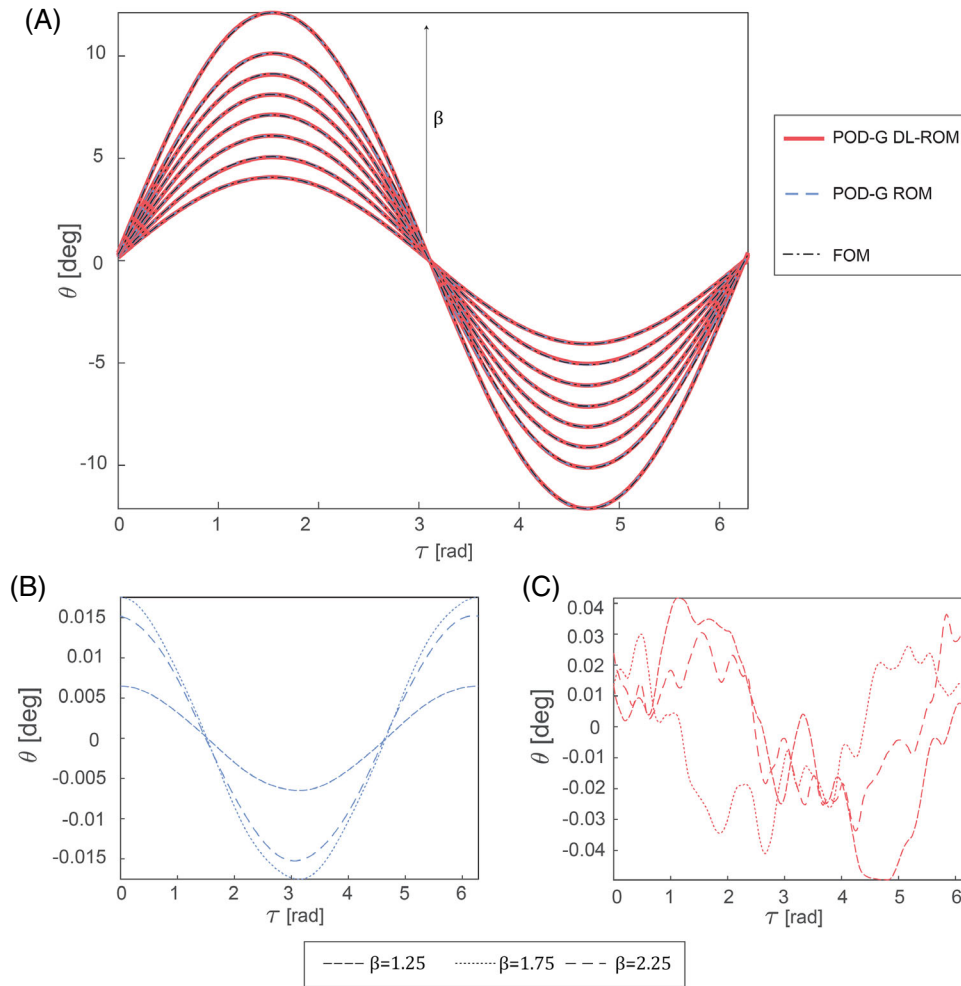


FIGURE 16 Micromirror. Comparison between the periodic responses of the FOM (black line), the POD-G ROM (blue line), and the POD-G DL-ROM (red line) solutions. The sampling points are indicated by black dots in Figure 14. (A) Evolution of the rotation angle over one period. (B, C) Angle error of the POD-G ROM and the POD-G DL-ROM solutions with respect to the FOM.

One of the major strengths of the approach is that, starting from the output $\tilde{\mathbf{u}}_N$ of the decoder, the FOM approximate solution can be reconstructed using the POD-G modes as: $\tilde{\mathbf{u}}_h(t; \boldsymbol{\mu}) = \mathbf{V}\tilde{\mathbf{u}}_N(t; \boldsymbol{\mu})$. This field can be later manipulated with standard FEM tools to generate any quantity of interest, like for example, stresses which are often required to monitor strength criteria.

This means that the same outputs as the original FOM can be delivered without limitations. It should be mentioned, however, that this might require to access the data structures of the FOM, like the connectivity matrix, inducing some overheads in the testing phase.

As an example, the Von Mises stress σ_{VM} is plotted in Figure 19 where the FOM solution is compared with the prediction of the POD-G DL-ROM at two different time instants in Figure 17. A quantitative inspection shows that an excellent accuracy is preserved. Some artefacts appear in the plots which are however associated with the rather coarse mesh utilized in the example.

Finally, we compare the computational cost of the FOM, the POD-G ROM and the POD-G DL-ROM solutions. The results are summarized in Table 6 and the computational cost trend is depicted in Figure 20. Contrary to the clamped-clamped beam example, the larger POD basis dimension (9 as opposed to 4) yields as expected an increase of the ratio $T_{\text{POD}}/T_{\text{DL}}$ which puts in better evidence the interest of the proposed approach over other model order reduction techniques for real-time applications.

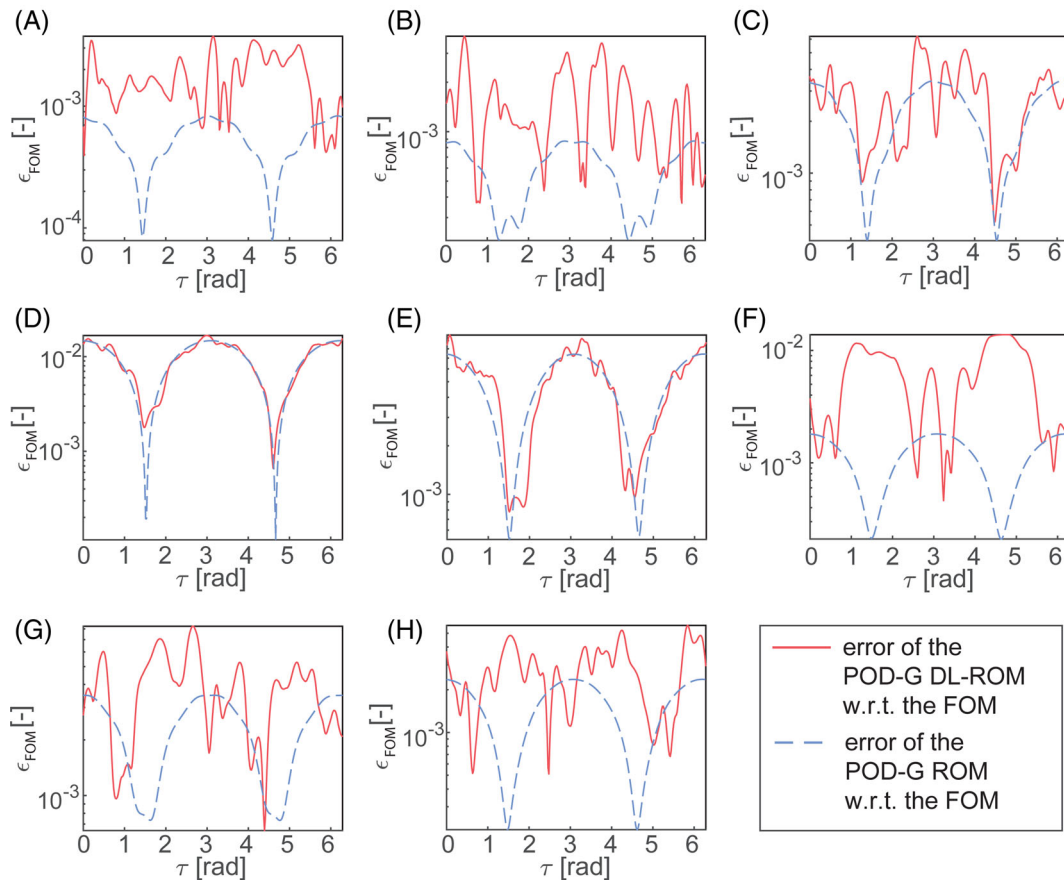


FIGURE 17 Micromirror. Relative error $\epsilon_{FOM}(t)$ (Equation 18), along one period, with respect to the FOM solution for the POD-G ROM (blue line) and the POD-G DL-ROM (red line) approximations. The sampling points are the ones detailed in Figure 14 and correspond to the peaks of the FRFs. (A) $\beta = 1.0$; (B) $\beta = 1.5$; (C) $\beta = 2.0$; (D) $\beta = 2.5$; (E) $\beta = 3.0$; (F) $\beta = 1.25$; (G) $\beta = 1.75$; (H) $\beta = 2.25$

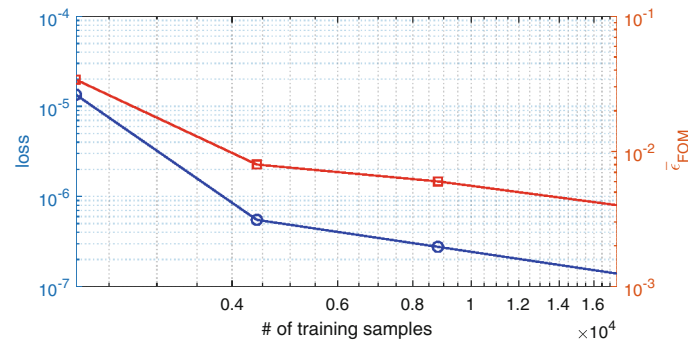


FIGURE 18 Micromirror. Convergence analysis with respect to the number of training samples. Two errors measures on testing instances are reported. The left axis refers to the loss (blue curve) while the right axis to the average relative error (red curve). The axes are in logarithmic scales.

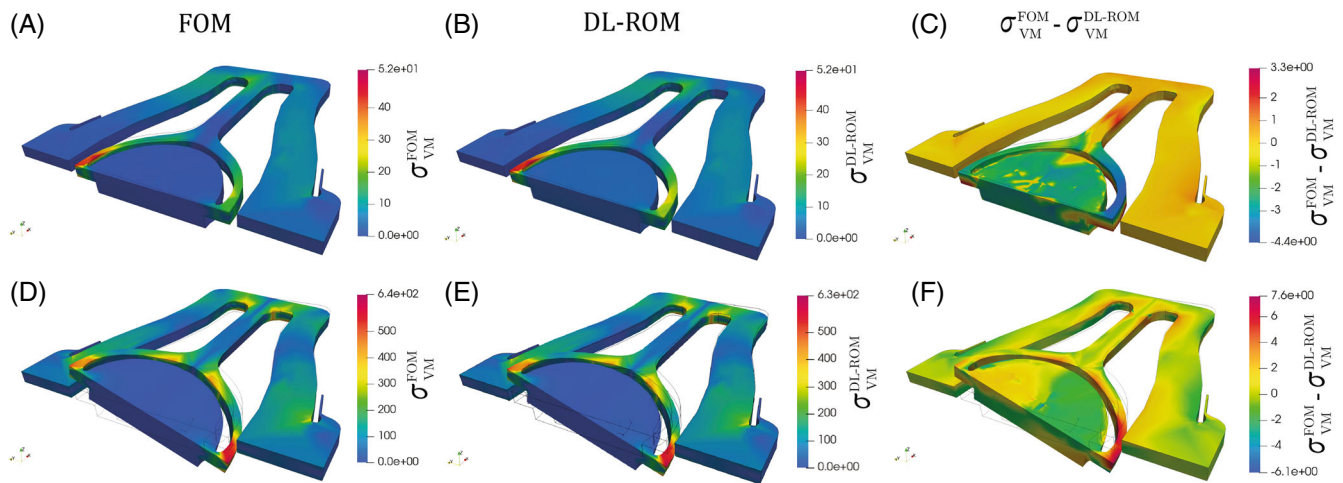


FIGURE 19 Contour plot of the Von Mises stress σ_{VM} at the peak of the FRF for $\beta = 2.25$ and two different instants of the periodic response that is, $\tau = \pi/2$ and $\tau = \pi$, with τ normalized time. (A, D) Field computed with the FOM. (B, E) Field computed with the POD-G DL-ROM. (C, F) Difference between the two reconstructions. The displacement of the mirror has been amplified by a factor 5.

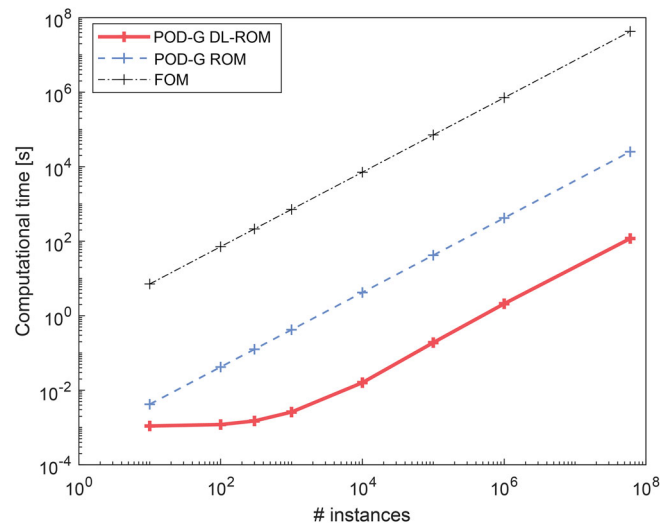


FIGURE 20 Micromirror. Plot of the time required to compute a given number of instances for the three solution level: FOM, POD-G ROM, and POD-G DL-ROM. All the methods display a linear trend. When a few instances are considered the POD-G DL-ROM displays a plate before the linear trend. This behavior is a consequence of the parallelization, under a certain threshold the number of cores overcomes the number of instances thus the time needed to get the result is nearly the same.

TABLE 6 Micromirror. Computational cost of the FOM (T_{FOM}), POD-G ROM (T_{POD}) and POD-G DL-ROM (T_{DL}) and speedup of the reduction techniques.

Number of instances	T_{FOM}	T_{POD}	T_{DL}	T_{FOM}/T_{DL}	T_{POD}/T_{DL}
10^1	7.09 s	0.0042	0.0011 s	6.4×10^3	3.81
10^2	1.18 m	0.042 s	0.0012 s	5.9×10^4	35
3×10^2	3.54 m	0.12 s	0.0015 s	1.41×10^5	80
10^3	11.82 m	0.42 s	0.0026 s	2.72×10^5	161
10^4	1.97 h	4.23 s	0.016 s	4.43×10^5	264
10^5	19.7 h	42.3 s	0.19 s	3.73×10^5	222
10^6	8.2 d	7 m	2.1 s	3.37×10^5	200
6×10^7	494 d	7.05 h	118 s	3.6×10^5	215

5 | CONCLUSIONS

In this article we have proposed a non-intrusive deep learning-based ROM tailored to describe both the trial manifold and the reduced dynamics of complex mechanical systems showing inertia and geometric nonlinearities. In order to reduce the computational cost of the procedure, a POD-Galerkin ROM has been first generated using a limited amount of FOM snapshots and the training of the DL-ROM has been performed using cheaper POD-G solutions covering the whole parameter range.

The encoder function of a convolutional autoencoder (AE) has been used to map the system response onto a low-dimensional representation, while the decoder part models the reduced nonlinear trial manifold and allows to reconstruct a posteriori an approximation of the FOM response. Finally, in order to describe the system dynamics on the manifold, a deep forward neural network has been utilized and trained at the same time as the AE.

The proposed technique has proved extremely efficient as it models highly nonlinear problems by identifying the manifold underlying the dynamics in a completely data-driven, black-box, and non-intrusive way, provided that the sampled data span the parameter space of interest.

This strategy has been implemented and benchmarked on both an academic example and a real industrial application like a MEMS micromirrors showing softening response and multiplicity of solutions. In both cases a truly real-time ROM is achieved that preserves the capability to retrieve a posteriori, through the decoder part of the AE, an estimate of any solution field available in the corresponding FOM.

In both applications, the POD-G DL-ROM has reproduced the dynamic behavior of the system with remarkable accuracy, inducing limited errors in the regions corresponding to the lower and upper bounds of the phase range. Concerning the computational cost, we have compared the POD-G DL-ROM both against the original FOM solved by HB procedures and against the POD-G ROM. In the former case the gain is always impressive, and in the latter, as soon as the POD trial-space dimension increases in realistic examples, improvements of two orders of magnitude have been observed.

The data generation phase through the POD-G ROM might however fail in complex scenarios like, for example, fully coupled fluid-structure simulations and in applications where the range of variation of the parameters is large. These situations require further investigation, but might be handled with the POD DL-ROM formulation in which the POD is only used to preprocess the matrix of snapshots to extract salient features retaining a large number of POMS.

ACKNOWLEDGMENT

Stefania Fresca, Giorgio Gobat, and Andrea Manzoni have been supported by Fondazione Cariplo, Grant no. 2019-4608. Open Access Funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

ORCID

Andrea Manzoni  <https://orcid.org/0000-0001-8277-2802>

REFERENCES

1. Shin DD, Chen Y, Flader IB, Kenny TW. Temperature compensation of resonant accelerometer via nonlinear operation; 2018:1012-1015; IEEE.
2. Defoort M, Taheri-Tehrani P, Horsley DA. Exploiting nonlinear amplitude-frequency dependence for temperature compensation in silicon micromechanical resonators. *Appl Phys Lett*. 2016;109(15):153502.
3. Agarwal M, Chandorkar SA, Candler RN, et al. Optimal drive condition for nonlinearity reduction in electrostatic microresonators. *Appl Phys Lett*. 2006;89(21):214105.
4. Rivlin B, Elata D. Design of nonlinear springs for attaining a linear response in gap-closing electrostatic actuators. *Int J Solids Struct*. 2012;49(26):3816-3822.
5. Chen C, Zanette DH, Czaplowski DA, Shaw S, López D. Direct observation of coherent energy transfer in nonlinear micromechanical oscillators. *Nature Commun*. 2017;8(1):1-7.
6. Sarrafan A, Azimi S, Golnaraghi F, Bahreyni B. A nonlinear rate microsensor utilising internal resonance. *Sci Rep*. 2019;9(1):1-9.
7. Rhoads JF, Shaw SW, Turner KL. Nonlinear dynamics and its applications in micro-and nanoresonators. *J Dyn Syst Measur Control*. 2010;132(3):034001.
8. Torteman B, Kessler Y, Liberzon A, Krylov S. Micro-beam resonator parametrically excited by electro-thermal Joule's heating and its use as a flow sensor. *Nonlinear Dyn*. 2019;98(4):3051-3065.
9. Zhang W, Turner KL. Application of parametric resonance amplification in a single-crystal silicon micro-oscillator based mass sensor. *Sens Actuat A Phys*. 2005;122(1):23-30.

10. Nitzan SH, Zega V, Li M, et al. Self-induced parametric amplification arising from nonlinear elastic coupling in a micromechanical resonating disk gyroscope. *Sci Rep*. 2015;5(1):1-6.
11. Polunin PM, Shaw SW. Self-induced parametric amplification in ring resonating gyroscopes. *Int J Non-Linear Mech*. 2017;94:300-308.
12. Bouchaala A, Jaber N, Yassine O, et al. Nonlinear-based MEMS sensors and active switches for gas detection. *Sensors*. 2016;16(6):758.
13. Frangi A, De Masi B, Confalonieri F, Zerbini S. Threshold shock sensor based on a bistable mechanism: design, modeling, and measurements. *J Microelectromech Syst*. 2015;24(6):2019-2026.
14. HaLevy O, Krakover N, Krylov S. Feasibility study of a resonant accelerometer with bistable electrostatically actuated cantilever as a sensing element. *Int J Non-Linear Mech*. 2020;118:103255.
15. Kessler Y, Krylov S, Liberzon A. Flow sensing by buckling monitoring of electrothermally actuated double-clamped micro beams. *Appl Phys Lett*. 2016;109(8):083503.
16. Kerschen G, Peeters M, Golinval J-C, Vakakis AF. Nonlinear normal modes, Part I: a useful framework for the structural dynamicist. *Mech Syst Signal Process*. 2009;23(1):170-194.
17. Renson L, Kerschen G, Cochelin B. Numerical computation of nonlinear normal modes in mechanical engineering. *J Sound Vib*. 2016;364:177-206.
18. Mignolet MP, Przekop A, Rizzi SA, Spottswood SM. A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures. *J Sound Vib*. 2013;332(10):2437-2460.
19. Besselink B, Tabak U, Lutowska A, et al. A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control. *J Sound Vib*. 2013;332(19):4403-4422.
20. Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. Springer; 2015.
21. Manzoni A, Negri F. *Efficient Reduction of PDEs Defined on Domains with Variable Shape*. Springer International Publishing; 2017:183-199.
22. Pagani S, Manzoni A, Quarteroni A. Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. *Comput Methods Appl Mech Eng*. 2018;340:530-558.
23. Farhat C, Grimmer S, Manzoni A, Quarteroni A. *Computational Bottlenecks for PROMs: Precomputation and Hyperreduction*. Vol 5. De Gruyter; 2020:181-244.
24. Touzé C, Amabili M. Nonlinear normal modes for damped geometrically nonlinear systems: application to reduced-order modelling of harmonically forced structures. *J Sound Vib*. 2006;298(4-5):958-981.
25. Ponsioen S, Pedergrana T, Haller G. Automated computation of autonomous spectral submanifolds for nonlinear modal analysis. *J Sound Vib*. 2018;420:269-295.
26. Opreni A, Vizzaccaro A, Frangi A, Touzé C. Model order reduction based on direct normal form: application to large finite element mems structures featuring internal resonance. *Nonlinear Dyn*. 2021;105:1237-1272.
27. Gobat G, Opreni A, Fresca S, Manzoni A, Frangi A. Reduced order modeling of nonlinear microstructures through proper orthogonal decomposition. *Mech Syst Signal Process*. 2022;171:108864.
28. Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput Methods Appl Mech Eng*. 2021;379:113741.
29. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys*. 2019;378:686-707.
30. Guo M, Hesthaven JS. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Comput Methods Appl Mech Eng*. 2018;341:807-826.
31. Guo M, Hesthaven JS. Data-driven reduced order modeling for time-dependent problems. *Comput Methods Appl Mech Eng*. 2019;345:75-99.
32. Lee K, Carlberg KT. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J Comput Phys*. 2020;404:108973.
33. Kaiser E, Kutz JN, Brunton SL. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc Royal Soc A*. 2018;474(2219):20180335.
34. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci*. 2016;113(15):3932-3937.
35. Brunton SL, Proctor JL, Kutz JN. Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine*. 2016;49(18):710-715.
36. Champion K, Lusch B, Kutz JN, Brunton SL. Data-driven discovery of coordinates and governing equations. *Proc Natl Acad Sci*. 2019;116(45):22445-22451.
37. Fresca S, Manzoni A, Dedè L, Quarteroni A. Deep learning-based reduced order models in cardiac electrophysiology. *PLoS One*. 2020;15(10):e0239416.
38. Fresca S, Dede L, Manzoni A. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. *J Sci Comput*. 2021;87(2):1-36.
39. Simpson T, Dervilis N, Chatzi E. Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks. *J Eng Mech*. 2021;147(10):04021061.
40. Wang Z, Xiao D, Fang F, Govindan R, Pain CC, Guo Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int J Numer Methods Fluids*. 2018;86(4):255-268.

41. Lui HFS, Wolf WR. Construction of reduced-order models for fluid flows using deep feedforward neural networks. *J Fluid Mech.* 2019;872:963-994.
42. Gonzalez FJ, Balajewicz M. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. arXiv preprint arXiv:1808.01346, 2018.
43. Amsallem D, Farhat C. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA J.* 2008;46(7):1803-1813.
44. Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *Int J Numer Methods Eng.* 2012;92(10):891-916.
45. Mosquera R, El Hamidi A, Hamdouni A, Falaize A. Generalization of the Neville–Aitken interpolation algorithm on Grassmann manifolds: applications to reduced order model. *Int J Numer Methods Fluids.* 2021;93(7):2421-2442.
46. Vizzaccaro A, Shen Y, Salles L, Blahoš J, Touzé C. Direct computation of nonlinear mapping via normal form for reduced-order models of finite element nonlinear structures. *Comput Methods Appl Mech Eng.* 2021;384:113957.
47. Vizzaccaro A, Opreni A, Salles L, Frangi A, Touzé C. High order direct parametrisation of invariant manifolds for model order reduction of finite element structures: application to large amplitude vibrations and uncovering of a folding point. arXiv preprint arXiv:2109.10031, 2021.
48. Malvern LE. *Introduction to the Mechanics of a Continuous Medium.* Prentice-Hall Series in Engineering of the Physical Sciences. Prentice-Hall; 1969.
49. Opreni A, Boni N, Carminati R, Frangi A. Analysis of the nonlinear response of piezo-micromirrors with the harmonic balance method. *Actuators.* 2021;10(21):1-12.
50. Detroux T, Renson L, Masset L, Kerschen G. The harmonic balance method for bifurcation analysis of large-scale nonlinear mechanical systems. *Comput Methods Appl Mech Eng.* 2015;296:18-38.
51. Kerschen G, Golinvall J-c, Vakakis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dyn.* 2005;41:147-169.
52. Lu K, Jin Y, Chen Y, et al. Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems. *Mech Syst Signal Process.* 2019;123:264-297.
53. Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Compt Rendus Math Acad Sci.* 2004;339(9):667-672.
54. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput.* 2010;32(5):2737-2764.
55. Maday Y, Nguyen N, Patera A, Pau G. A general multipurpose interpolation procedure: the magic points. *Commun Pure Appl Anal.* 2008;8:383-404.
56. Fresca S, Manzoni A. POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Comput Methods Appl Mech Eng.* 2022;388:114181.
57. Halko N, Martinsson P, Tropp JA. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* 2011;53:217-288.
58. Doedel EJ, Champneys AR, Dercole F, et al. AUTO-07P: continuation and bifurcation software for ordinary differential equations; 2007.
59. Guillot L, Cochelin B, Vergez C. A Taylor series-based continuation method for solutions of dynamical systems. *Nonlinear Dyn.* 2019;98(4):2827-2845.
60. Krack M, Gross J. *Harmonic Balance for Nonlinear Vibration Problems.* Springer; 2019.
61. Veltz R. BifurcationKit.jl; 2020. HAL: <https://hal.inria.fr/hal-02902346>
62. Thomsen JJ, Thomsen JJ, Thomsen JJ. *Vibrations and Stability.* Springer; 2003.
63. Zega V, Gattere G, Koppaka S, et al. Numerical modelling of non-linearities in MEMS resonators. *J Microelectromech Syst.* 2020;29(6):1443-1454.
64. Mestrom RMC, Fey RHB, Nijmeijer H. Phase feedback for nonlinear MEM resonators in oscillator circuits. *IEEE/ASME Trans Mechatron.* 2009;14(4):423-433.
65. Kingma DP, Ba J. *Adam: A Method for Stochastic Optimization.* 2014;arXiv:1412.6980.
66. Goodfellow I, Bengio Y, Courville A. *Deep Learning.* MIT Press; 2016.
67. Clevert D, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289, 2015.
68. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2015:1026-1034.
69. Abadi M, Barham P, Chen J, et al. TensorFlow: a system for large-scale machine learning; 2016:265-283.
70. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the JMLR Workshop and Conference; 2010:249-256.
71. Corigliano A, De Masi B, Frangi A, Comi C, Villa A, Marchi M. Mechanical characterization of polysilicon through on-chip tensile tests. *J Microelectromech Syst.* 2004;13(2):200-219.
72. Laser beam scanning. https://www.st.com/content/st_com/en/about/innovation---technology/laser-beam-scanning.html
73. Frangi A, Opreni A, Boni N, et al. Nonlinear response of PZT-actuated resonant micromirrors. *J Microelectromech Syst.* 2020;29(6):1421-1430.

74. Frangi A, Gobat G. Reduced order modelling of the non-linear stiffness in MEMS resonators. *Int J Non-Linear Mech.* 2019;116: 211-218.

How to cite this article: Fresca S, Gobat G, Fedeli P, Frangi A, Manzoni A. Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures. *Int J Numer Methods Eng.* 2022;123(20):4749-4777. doi: 10.1002/nme.7054