EMERGING TECH CONFERENCE – Edge Intelligence

Volume 01, 2022, Page 15 – 25

Proceedings of Emerging Tech Conference: Edge Intelligence 2022

An Artificial Intelligence Framework for Non-Intrusive Reduced Order Modeling

George Drakoulas[a,*], Theodoros Gortsas[a], Charilaos Kokkinos[b], Kopsaftopoulos Fotis[c], Demosthenes Polyzos[a]

[a]*Department of Mechanical Engineering and Aeronautics, University of Patras, GR-26500, Patras, Greece*
[b]*FEAC Engineering P.C., GR-26224, Patras, Greece*
[c]*Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Insitute, NY 12180, Troy, USA*

g.drakoulas@upnet.gr

Abstract

Reduced order models (ROMs) are employed, to approximate the high-fidelity solutions while accurately capturing the dominant aspects of the physical behavior. In this work, a non-intrusive ROM is employed to reduce the computational cost and provide numerical solutions in a limited amount of time. The utilized framework, mentioned as FastSVD-ML-ROM applies (i) a singular value decomposition (SVD) update methodology to compute a linear subspace of the high-fidelity solutions during the simulation process, (ii) convolutional autoencoders for nonlinear dimensionality reduction, (iii) feed-forward neural networks to map the input parameters to the latent spaces, and (iv) long short-term memory networks to predict the dynamics of parametric solutions. The efficiency of the FastSVD-ML-ROM is demonstrated both for the Navier stokes equation in the frame of fluid dynamics and for the Navier-Cauchy equations dealing with structural dynamics. In particular, the developed framework is used to monitor the flow field around a cylinder in real-time and for the wave propagation on aluminum plates targeting to evaluate the effect of the uncertainties introduced in the material properties. The accuracy of the predicted results compared to the high-fidelity solutions showcase the robustness of the proposed approach.

1. Introduction

High-fidelity modes (HFMs) in terms of numerical analysis are commonly used to solve complex physical phenomena and provide a great insight into the performance of the system before its deployment in the real world. Amongst others, computational fluid dynamics and computational structural mechanics are applied in the field of aerospace, bio-engineering, marine, and energy to simulate with great accuracy, non-linear, time-dependent partial differential equations (PDEs) [1, 2, 3].

Although HFMs constitute the foundation of engineering design, the techniques used in the fourth industrial revolution to construct and monitor large structures require numerical models that can provide solutions in a limited amount of time [4]. In addition, modern technologies including edge computing, cloud services (e.g. Azure), and computer hardware, such as multi-core graphics processing units, offer excellent potential for the further development of data-driven approaches [5]. Besides, various methods of machine learning have been employed to uncover hidden patterns in noisy data [6].

Along these lines, surrogate modeling has gained attention aiming to provide physics-based solutions, accurately, in real-time, while decreasing the computational time and cost. Reduced order models (ROMs) are a mathematical approach, part of surrogate modeling [7]. ROMs can efficiently map large-scale numerical data in a low-dimensional space. In recent years, machine learning (ML)-based ROMs are utilized to eliminate the computational cost of high-fidelity solutions by recovering non-linear, low-dimensional manifolds, while capturing the behavior of unsteady, complex, multi-physics simulations [8].

In this work, a generic ML-based ROM framework presented by Drakoulas et. al. [9], referred to as *FastSVD-ML-ROM* is utilized. The applied framework includes a singular value decomposition (SVD) update methodology, during the simulation process to handle big snapshot matrices, and identify low-dimensional embeddings. Then, the linear projected data is passed to the convolutional autoencoders (CAEs) for non-linear dimensionality reduction. Moreover, long short-term memory (LSTM) network is employed to predict the temporal scales of the reduced representations. Then, feed-forward neural networks (FFNNs) are utilized during the online phase to map the input parameters to the latent representations.

The *FastSVD-ML-ROM* is applied for two test cases: *i)* the 2D fluid flow around a cylinder (Navier-stokes equation) *ii)* and 3D guided wave propagation (Navier- Cauchy equation). In the first test case, we aim to monitor the flow field around a cylinder, and the objective of the second test case is to estimate the uncertainties introduced in the material properties of an aluminum plate.

The structure of this paper is as follows. Section 2, presents the methodology of the *FastSVD-ML-ROM*, including the SVD update methodology, and the CAEs for the non-linear dimensionality reduction, as well. Moreover, the combination of the LSTMs and the FFNNs utilized for the prediction of the dynamics is illustrated. Then in section 3, the implementation of the developed framework for the *i)* flow past a cylinder and the *ii)* wave propagation on the plate is presented. Finally, a brief discussion of the results is described in section 4.

2. Machine learning-based Reduced Order Modeling

2.1. Snapshot matrix construction

We consider a time-dependent, parametrized PDE, with an unknown field $u : R^{+} \times R^{d} \times R^{\xi} \to R^{g}$ where ξ is the dimension of the parameter vector, d is the number of spatial dimensions. After the spatial discretization of the PDE, we uniformly sample μ parameters from the space P and split into two sets, $\boldsymbol{\mu}^{tr} = \{\boldsymbol{\mu}_1^{tr}, \dots, \boldsymbol{\mu}_m^{tr}\}$ and $\boldsymbol{\mu}^{te}$ $\{\boldsymbol{\mu}_1^{te}, \dots, \boldsymbol{\mu}_m^{te}\}$ corresponding to the m training and n testing parameters, respectively.

The snapshot matrix $\boldsymbol{S}_h$ is then defined for the training parameter set $\boldsymbol{\mu}^{tr}$ and the discrete time set with, as follows,

$$S_h = [\boldsymbol{u}_h(t_1 \boldsymbol{\mu}_1^{tr})| \dots |\boldsymbol{u}_h(t_{N_t}, \boldsymbol{\mu}_1^{tr}) \dots |\boldsymbol{u}_h(t_1, \boldsymbol{\mu}_m^{tr})| \dots |\boldsymbol{u}_h(t_{N_t}, \boldsymbol{\mu}_m^{tr})] \quad (1)$$

where $\boldsymbol{u}_h(t_i, \boldsymbol{\mu}_j^{tr})$ is a snapshot of the parameter $\boldsymbol{\mu}_j^{tr}$ at time t_i.

2.2. Dimensionality reduction

The *FastSVD-ML-ROM* utilizes a hyper-reduction technique based on both linear algebra (SVD) update methodology and CAE for the non-linear compression of the data. Primary, the truncated SVD update is employed during the simulation process, to obtain an approximate basis vector $\mathbf{U}$ from the snapshot matrix (Eq. 1). Then using the calculated basis, the reduced data is obtained as,

$$\tilde{\mathbf{u}}_h(t_i; \boldsymbol{\mu}_j^{tr}) \approx \mathbf{U}\ \mathbf{U}^T \mathbf{u}_h(t_i; \boldsymbol{\mu}_j^{tr}) \tag{2}$$

Given the snapshot matrix containing the solutions for all the parameters, we calculate the following low-dimensional representation,

$$\mathbf{S}_n = \mathbf{U}^T \mathbf{S}_h \tag{3}$$

where the projected matrix $\mathbf{S}_n$ is the input of the CAE and n the column dimensions of the basis vector $\mathbf{U}$ derived during the SVD update. Therefore, the projected FOM solution for the training parameter $\boldsymbol{\mu}_j^{tr}$ at time t_i is derived through the formula,

$$\mathbf{u_n}(t_i, \boldsymbol{\mu}_j^{tr}) = \mathbf{U}^T \mathbf{u_h}(t_i; \boldsymbol{\mu}_j^{tr}) \tag{4}$$

As a second step, the projected solutions $\mathbf{u_n}(t_i, \boldsymbol{\mu}_j^{tr})$, with j=1, … , m and i=1, … , N_t, are given as input to the CAE, aiming to identify non-linear, low-dimensional, latent variables.

Let $\boldsymbol{\phi}_e$ be the encoder function that attempts to discover a latent representation $\tilde{\mathbf{z}}_q \in \mathbb{R}^q$, with $q << n$ as,

$$\tilde{\mathbf{z}}_\mathbf{q}\big(t_i; \boldsymbol{\mu}_j^{tr}\big) = \boldsymbol{\phi}_e\big(\mathbf{u}_n\big(t_i; \boldsymbol{\mu}_j^{tr}\big); \mathbf{W}_e; \mathbf{b}_e\big) \tag{5}$$

$with\ \ i = 1, \dots, N_t\ \ and\ \ j = 1, \dots, m$ while $\mathbf{W}_e$ $\mathbf{b}_e$ are groups of weight matrices and bias vectors of the encoder, respectively. The decoder reconstructs the approximate input data $\tilde{\mathbf{u}}_\mathbf{n}(t_i; \boldsymbol{\mu}_j^{tr})$ through the following transformation,

$$\tilde{\mathbf{u}}_n\big(t_i; \boldsymbol{\mu}_j^{tr}\big) = \boldsymbol{\phi}_d\big(\tilde{\mathbf{z}}_q\big(t_i; \boldsymbol{\mu}_j^{tr}\big); \mathbf{W}_d; \mathbf{b}_d\big) \tag{6}$$

where the transition functions are defined as $\boldsymbol{\phi}_e: \mathbb{R}^n \to \mathbb{R}^q$ and $\boldsymbol{\phi}_d: \mathbb{R}^q \to \mathbb{R}^n$, while $\mathbf{W}_d$, $\mathbf{b}_d$ are the weight matrices and bias vectors of the decoder.

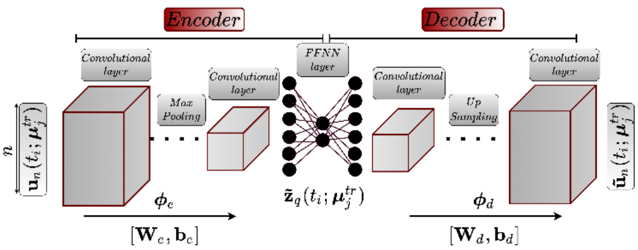

Figure 1: CAE architecture including the encoder and decoder parts.

2.3. Time prediction

In the frame of *FastSVD-ML-ROM*, to predict the spatio-temporal solutions of the reduced data $\tilde{\mathbf{z}}_\mathbf{q}$, which is extracted from the dimensionality reduction (Section 2.2), a FFNN and LSTM network is utilized. In particular, the DL models are trained independently over the training and then during the online phase, the FFNN predicts and provides the first time series to the LSTM network, used to forecast the temporal solutions.

The LSTM network denoted as l takes as inputs the vector $\mathbf{x}_l^i(\boldsymbol{\mu}_j^{tr})$, consisting of the latent spaces $\tilde{\mathbf{z}}_q(\boldsymbol{t}_i, \boldsymbol{\mu}_j)$ with $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{N}_t$ and $\boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{m}$ extracted through the trained encoder $\boldsymbol{\varphi}_e$. To explicitly account for the parameter information, we concatenate the latent spaces with the parameter values, and thus $\mathbf{x}_l^i(\boldsymbol{\mu}_j^{tr}) = [\tilde{\mathbf{z}}_q(\boldsymbol{t}_i; \boldsymbol{\mu}_j^{tr}), \boldsymbol{\mu}_j^{tr}]$.

The LSTM network is applied sequentially for each training parameter set $\boldsymbol{\mu}_j^{tr}$ at time $\boldsymbol{t}_i$ and mainly consists of three gates acting as filters for the input data. The forget gate $\mathbf{f}_i(\boldsymbol{\mu}_j^{tr})$ that decides which information is less important and can be ignored, the input gate $\mathbf{i}_i(\boldsymbol{\mu}_j^{tr})$ used to update the cell state and the output gate $\mathbf{o}_i(\boldsymbol{\mu}_j^{tr})$ determines the information that will be passed to the next state. Besides, the updated cell state $\mathbf{C}_i(\boldsymbol{\mu}_j^{tr})$, composed of the input $\mathbf{i}_i(\boldsymbol{\mu}_j^{tr})$, the forget gate $\mathbf{f}_i(\boldsymbol{\mu}_j^{tr})$ and the candidate's memory state $\tilde{\mathbf{C}}_i(\boldsymbol{\mu}_j^{tr})$ generates data. The hidden state $\mathbf{h}_i(\boldsymbol{\mu}_j^{tr})$ produces the output data given the filtered data from the output gate $\mathbf{o}_i(\boldsymbol{\mu}_j^{tr})$. Then, the hidden state is passed to a FFNN layer aiming to predict the final output $\tilde{\mathbf{y}}_i\ (\boldsymbol{\mu}_j^{tr})$.

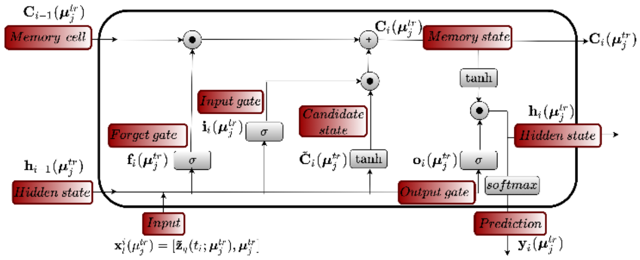

Figure 2: Architecture of the LSTM network, as part of the *FastSVD-ML-ROM*.

2.4. Parameter regression

During the offline phase, along with the training of the LSTMs, FFNNs are used to identify the latent space vectors $\tilde{\mathbf{z}}_\mathbf{q}(\boldsymbol{t}_i; \boldsymbol{\mu}_j^{tr})$ for $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{w}$ and $\boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{m}$ that form the first time series matrix, provided as input to the LSTM during the online phase to predict and forecast the dynamics. To formulate the input for the FFNNs, we concatenate the time instances $\boldsymbol{t}_i$, $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{w}$ with the parameters $\boldsymbol{\mu}_j^{tr}$, and thus $\mathbf{x}_f^i(\boldsymbol{\mu}_j^{tr}) = [\boldsymbol{t}_i, \boldsymbol{\mu}_j^{tr}]$.

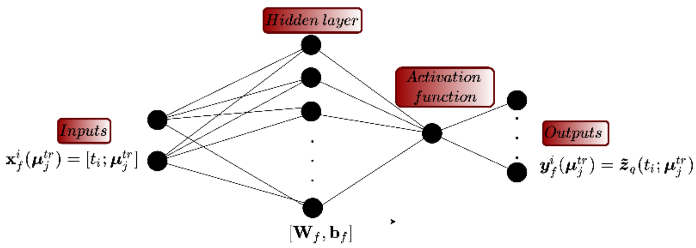

Figure 3: The architecture of the FFNN, as part of the *FastSVD-ML-ROM*.

3. Numerical results

Two benchmark problems are employed to assess the performance of the proposed *FastSVD-ML-ROM* framework, namely, *i)* the flow around a cylinder (section 3.1) and *ii)* the wave propagation on an aluminum plate (section 3.2). The DL models are implemented in TensorFlow 2.3 and the SVD technique is executed by the NumPy library in python 3.8.3. The numerical simulations and the training of the DL models are executed on an Intel Core i7 @ 2.20GHz, NVIDIA GeForce GTX 1050 Ti GPU personal computer.

3.1. Test case 1: Flow past a cylinder

In this example, the unbounded external flow around a cylinder with a circular cross-section is examined. We numerically solve the incompressible Navier-Stokes (N-S) equations—in their primitive variables (velocity ***u*** and pressure *p*) formulation— written as:

$$\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \nabla \cdot 2v\boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{f} \tag{7}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}$$

The term $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor defined as:

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right) \tag{8}$$

while $\mathbf{f}$ are the body forces and v is the kinematic viscosity of the fluid.

The flow domain is large compared to the dimensions of the cylinder as shown in Fig. 4. The center of the 0.01m in diameter cylinder, is located at point $O(0.1m,\ 0.1m)$ of a square domain with dimensions $0.0 \le x \le 0.5$ and $0.0 \le y \le 0.2$. The flow is not perturbed near the inlet, which is located far from the cylinder. The inflow is defined from the left side and the outflow from the right side, respectively. No-slip conditions are defined on the wall and the cylinder. The flow domain is discretized with 211,000 finite volume cells, locally refined in the vicinity of the cylinder. The grid resolution ensures a grid-independent numerical solution. We numerically solve the governing equations using the commercial software Siemens Star CCM+ [10]. The fluid's density is ρ= 997 kg/m^3 and the kinematic viscosity is v = 1.0034 mm^2/s.

The main input parameter that formulates the snapshot matrix is the Reynolds number (Re). We keep the indicated region that consists of the area of interest, including 13,834 cells (Fig. 4). We consider the Reynolds number as the parameter, in the space P = [100, 210]. This range is chosen to generate non-linear flow patterns, e.g. Karman vortex streets behind the cylinder. We uniformly sample m = 9, training, and n = 2 testing parameters, as well. The unsteady case is solved with the non-iterative PISO algorithm, over the interval t = [0, 120] s, with a time-step $dt = 5 \times 10^{-4}$ s resulting in 240,000 snapshots. We then choose 480 solutions, with a step equal to 500, and we keep the last 260 instances, belonging to the time range t = [55, 120] s. We implement the *FastSVD-ML-ROM*, to reconstruct the velocity magnitude field in real-time, during the online phase, for a given testing parameter set.

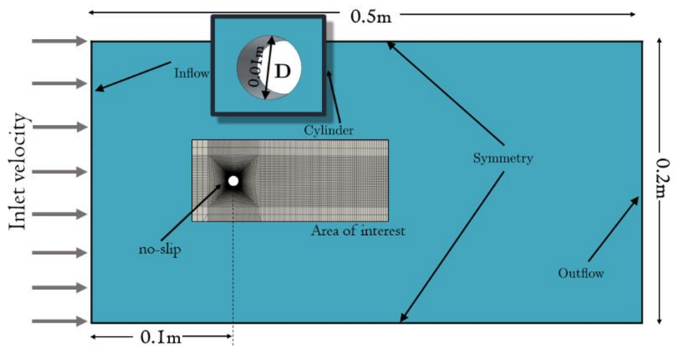

Figure 4: The geometry and the boundary conditions of the flow around a cylinder case.

In Fig. 5, to access the accuracy of the FFNN and the LSTM predictions, we present the evolution of the latent space representations along with time for various training parameter sets **{**130, 150, 185, 210}. It is observed that the FFNN can precisely capture the first sequential data containing the latent space representations until t = 65 s. Moreover, the LSTM can accurately predict the time evolution of the latent space representations for the entire time range (t = [120, 210] s). The combination of the FFNN and the LSTM can capture the dynamics of the latent space representations extracted from the trained encoder, part of the CAE.

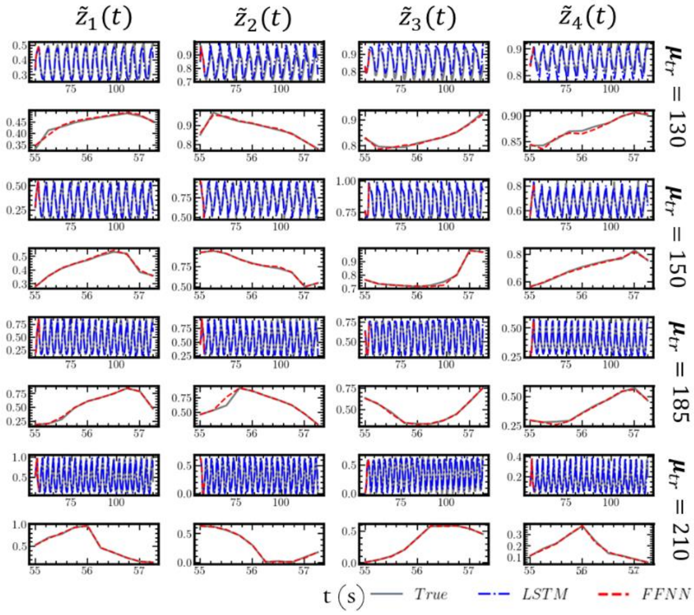

Figure 5: True latent variables extracted by the trained decoder, LSTM, and FNNN predictions.

In Fig. 6, larger absolute values are obtained in the area behind and close to the cylinder, where the von-Karman vortices are starting to form. In Fig. 6, we demonstrate the ability of the surrogate model to forecast, in unseen times, t > 120 s. The velocity magnitude on the point behind the cylinder and the *erel*. along with time are depicted in Fig. 6 (bottom-left) plot while the evolution of the relative error *erel* along with time is presented in Fig. 6 (bottom-right). It is shown that the LSTM network can accurately capture the dynamics, enabling an additional 100% forecast, starting from the predicted solutions with a decreased relative error *erel*.

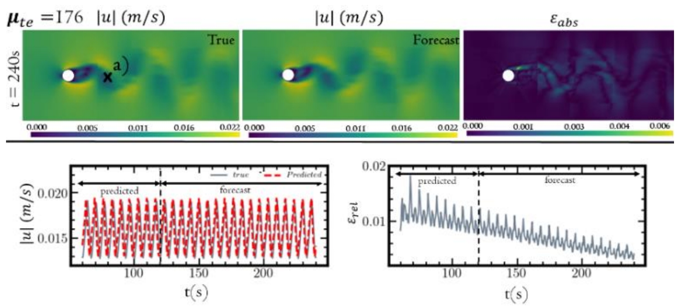

Figure 6: Comparison of the true and forecast velocity (top) and tracing on a point a (x), behind the cylinder and e_{rel} versus time (bottom).

The accuracy of the proposed methodology for the flow around a cylinder case is presented in Fig. 7. It is shown that the figures of the predicted velocity field are in good agreement with the velocity field obtained by the HFM and the ROM and can capture the vortices along the cylinder wake. The periodic pattern of the vortex shedding is observed both for the testing 200 and training 176 parameters.

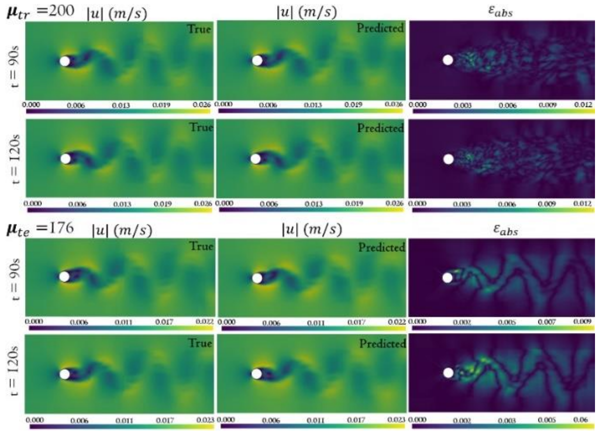

Figure 7: Comparison of the true and predicted velocity fields.

3.2. Test case 2: Wave propagation on an aluminum plate

The propagation of a lamb wave is simulated in terms of numerical analysis using the finite element method. The snapshot matrix is constructed through the variation of the material properties assuming Gaussian distributions. Experiments are performed on a rectangular aluminum plate under ambient temperature (Fig. 8). The numerical model is implemented by utilizing the SIEMENS Simcenter NX Nastran solver [11]. The transient plane stress structural problem is considered to simulate the wake propagation on the plate. The problem is governed by the equation of motion in 3-D linear elasticity as,

$$\frac{\partial^2 u_i}{\partial t^2} = \sum_{k=1}^{3} \left[\frac{E}{2\rho(1+\nu)} \frac{\partial^2 u_i}{\partial x_k \partial x_k} + \frac{E}{2\rho(1-\nu)} \frac{\partial^2 u_k}{\partial x_i \partial x_k} \right], \qquad i = 1,2,3 \tag{9}$$

where u_i= u_i (x,y,z), i=1,2,3 are the displacement fields in the x-, y- and z-axis respectively, E is the Young's modulus, ρ is the density and v is the Poisson's ratio, while the body forces are assumed to be zero.

Since the plate is a thin-wall structure, and the ratio of the thickness to the in-plane dimension is high, quadrilateral, isoparametric, and plane-strain elements are utilized. In particular, a global mesh size of 0.75mm is derived through a mesh convergence study, resulting in 82,028 nodes. A five-peak tone burst signal with a center frequency of 250 kHz is used to excite the plate in the z-axis direction while the left and right edges of the plate are kept fixed as shown in Fig. 8. Regarding, the set of the transient structural dynamics for the wave propagation a total time T=0.0001 s, is defined with a timestep dt=$10^{-7}s$, resulting in 1000 timesteps. We then choose 200 solutions, with a step equal to 5.

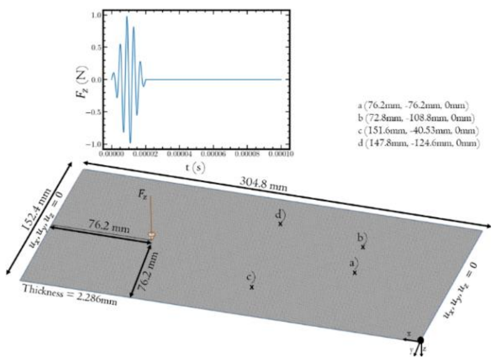

Figure 8: Geometry and mesh grid of the rectangular plate including the boundary conditions.

To numerically simulate the variations in the material properties on the propagation of lamb waves, we consider three Gaussian probability distributions for the Young's modulus (E), the density (ρ), and the Poisson's ratio (v), as well. The mean value of the Young's modulus is taken to be 68.9 GPa with a standard deviation of 1.332 GPa. The mean value of Poisson's ratio is 0.33 with a standard deviation of 0.007. As concerns the density, a mean value of 2700 kg/m^3 is selected including a standard deviation of 2.7 kg/m^3 [12]. Each parameter set vector used to solve Eq. and generate the HFM solutions is formulated by the triple {[E, v, ρ]}. In particular, the governing equation is solved for m=24 training parameter sets, while n=2 testing parameters are utilized to test the *FastSVD-ML-ROM*.

To gain a better insight into the ability of the *FastSVD-ML-ROM*, to capture the dynamics, we examine the performance of the LSTM and FFNN (Fig. 9). This is achieved by comparing the extracted latent variables from the trained encoder with the FFNN and LSTM predicted solution versus time. It is shown that the FFNN can accurately predict the first *w* components of the temporal evolutions, corresponding to the time interval t = [0, 0.2x10^{-5}] s, required by the LSTM to predict the entire evolution in time during the online phase, t = [0, 10^{-4}] s. It is also evident that the LSTM can precisely capture the dynamic behavior until t = 10^{-4} s.

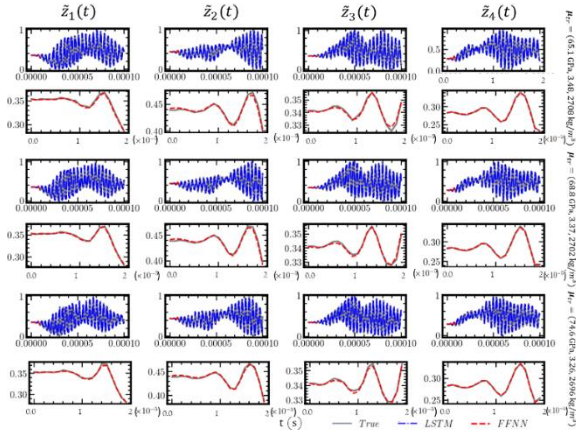

Figure 9: True latent variables extracted by the trained decoder, LSTM, and FNNN predictions.

The results of true and predicted results are presented in Fig. 10. We note the ability of the *FastSVD-ML-ROM* to accurately predict the acceleration distribution on the plate.

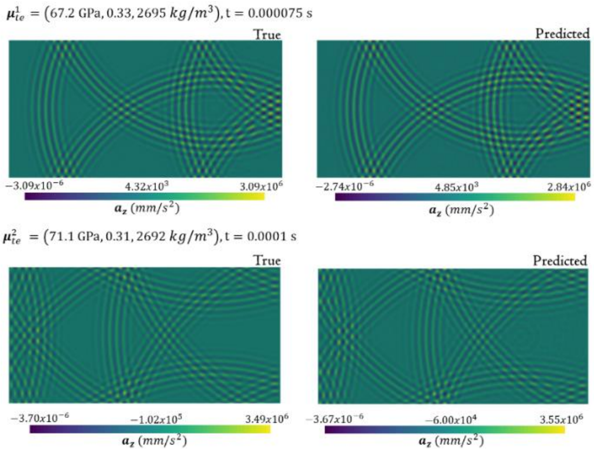

Figure 10: Comparison between the true and predicted solutions of the acceleration fields.

It is obvious that the *FastSVD-ML-ROM* can accurately monitor the acceleration values in the z-axis near and far from the excitation location and effectively model the time-varying (non)linear dynamics, including complex wave interaction phenomena. In Fig. 11 (upper-left), by examining the temporal evolution on node *a*, the symmetric S_0 and anti-symmetric A_0 modes are evident. In Fig. 11, (upper-left), the S_0 mode is observed while the A_0 seems to be mixed caused by the reflections from the edge. As concerns the temporal evolution of the node c (Fig.11, bottom-left), it is shown that the S_0 and A_0 modes are observed, followed by another wave packet while the temporal evolution on node *d* (Fig. 11, bottom-right) presents an overlapping with another wave packet.

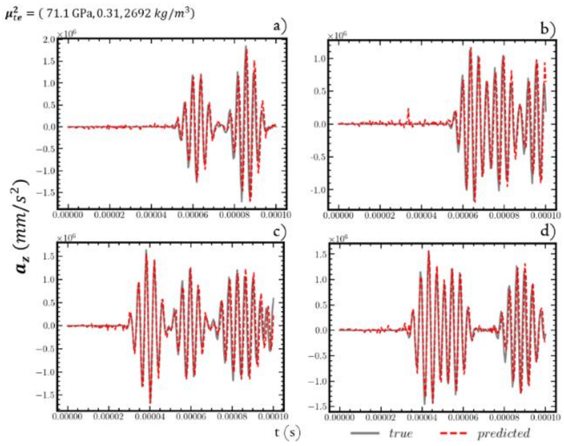

Figure 11: Monitoring of the true and predicted acceleration solutions on nodes a,b,c, and d.

Once the proposed ROM framework, is trained and tested, the uncertainty quantification caused due to the variation introduced into the material properties is examined. To gain a better insight into the effect of the young's modulus E, Poisson's ratio v, and density ρ we sample 1000 parameters for the UQ investigation.

Fig. 12, presents the UQ plots for the 1000 different signals on nodes a,b,c, and d (Fig. 8), generated by considering the variation in the material properties. The black lines represent the mean value of the time series while the gray area represents the +-1 standard deviation of the data. It is observed that the variation in Young's modulus, the Poisson's ratio, and the density of the aluminum plate, sampled from Gaussian distributions significantly affects the acceleration evolution in the z-axis (excitation direction). In particular, it appears that the change in the material properties, mainly affects the amplitude of the signals, with slight variations in their phase.

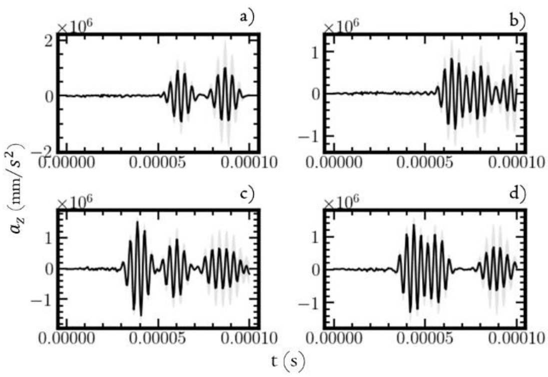

Figure 12: Mean acceleration (black line) and +-1 standard deviation distribution (gray area) on nodes a, b, c, and d.

4. Conclusions

In this paper, we utilize a NIROM framework, mentioned as *FastSVD-ML-ROM* to replace the computationally costly HFMs and provide accurate solutions for transient (non)-linear physics-based simulations in real-time. The employed surrogate models aim to build a virtual replica of the physical asset. The comparison between the HFM and ROM solutions indicates the robustness and efficiency of the applied methodology. As presented, the ROM reconstructions provide accurate results compared with the numerical data, while providing a tremendous speedup.

In particular, we present two benchmark problems dealing with, *i)* the 2D fluid flow around a cylinder solved by the navier-stokes equation and, *ii)* the guided wave propagation on an aluminum plate simulated with the navier-cauchy equations. The comparison among the high fidelity solutions and the ROM reconstructions demonstrates the ability of the *FastSVD-ML-ROM* to accurately predict the unsteady fields of interest in a limited amount of time, for a given parameter vector. As concern the fluid dynamics test case, the ROM achieves a maximum speed up $1\text{x}10^5$ compared to STAR-CCM+, and regarding the case of the structural dynamics, the ROM reconstructions provide a speed-up of $5\text{x}10^5$, in contrast to the NX Nastran solver.

Acknowledgments

The present work is supported by the Karlsruhe Institute of Technology under the project 81227 entitled «DAAD-Development of teaching for the installation and operation of wind turbines computer-aided modeling».

References

[1] G. Drakoulas, C. Kokkinos, D. Fotiadis, S. Kokkinos, K. Loukas, A. Moulas, A. Semertzioglou, Coupled fea model with continuum damage mechanics for the degradation of polymer-based coatings on drug-eluting stents, in: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, 2021, pp. 4319–4323.

[2] Gortsas, Theodore V., Stephanos V. Tsinopoulos, and Demosthenes Polyzos. "An accelerated boundary element method via cross approximation of integral kernels for large-scale cathodic protection problems." Computer-Aided Civil and Infrastructure Engineering 37.7 (2022): 848-863.

[3] Pettas, D., et al. "Stability analysis of a Newtonian film flow over hydrophobic microtextured substrates." Physical Review Fluids 7.3 (2022): 034004.

[4] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the digital twin: A systematic literature review, CIRP Journal of Manufacturing Science and Technology 29 (2020) 36–52.

[5] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers, arXiv preprint arXiv: 1910.01719 (2019).

[6] G. C. Peng, M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, et al., Multiscale modelling meets machine learning: What can we learn?, Archives of Computational Methods in Engineering 28 (2021) 1017–1037.

[7] F. J. Gonzalez, M. Balajewicz, Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, arXiv preprint arXiv:1808.01346 (2018).

[8] R. Swischuk, L. Mainini, B. Peherstorfer, K. Willcox, Projection-based model reduction: Formulations for physics-based machine learning, Computers & Fluids 179 (2019) 704–717

[9] G. I. Drakoulas, T. V. Gortsas, G. C. Bourantas, V. N. Burganos, and D. Polyzos, "FastSVD-ML-

ROM: A reduced-order modeling framework based on machine learning for real-time applications," arXiv preprint arXiv:2207.11842, 2022
[10] U. Guide, Starccm+ version 2020.1, SIEMENS simcenter (2020).
[11] Nastran, N. X. "Siemens PLM Software." Plano, TX, USA.
[12] S . Ahmed and F. Kopsaftopoulos, "Uncertainty quantification of guided waves propagation for active sensing structural health monitoring," in Proceedings of the vertical flight society 75th annual forum & technology display, Philadelphia, PA, USA, pp. 13–16, 2019.