12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Reduced order models of centrifugal pump for control applications: a comparison of Galerkin-projection and neural networks ⋆

Ali Mjalled*, Kamil Sommer*, Yogesh Parry Ravichandran, Romuald Skoda**, Martin Mönnigmann***

* *Autom. Contr. & Syst. Theory, Ruhr-Universität Bochum, Germany.*
** *Hydraulic Fluid Machinery, Ruhr-Universität Bochum, Germany.*
(e-mail: ali.mjalled@ruhr-uni-bochum.de)

Abstract: Galerkin-projection and non-intrusive neural network reduced order models (ROMs) for a two-dimensional centrifugal pump model are presented and compared with respect to their suitability for control purposes. Singular value decomposition (SVD) is applied to reduce the dimensionality of the model and to extract a set of reduced basis functions. In Galerkin-projection, the temporal evolution of the reduced coefficients is expressed in terms of a small set of ordinary differential equations (in the order of ten). To improve the performance of this method, we fit the operators obtained by the projection step to the original data. On the other hand, the regression step of the coefficients is performed using a deep recurrent neural network (RNN) in the non-intrusive method. We compare the two methods with respect to the computational time required to build and evaluate each model, relative prediction error, long-term stability and sensitivity to initial conditions. Our results show that the projection-based ROM is slightly more accurate than the non-intrusive ROM, but it requires more time to be built. However, the non-intrusive ROM is more stable and less sensitive to initial condition variation.

Keywords: Reduced order model, neural networks, singular value decomposition, Galerkin projection, long short-term memory, incompressible Navier-Stokes equation.

1. INTRODUCTION

Computational fluid dynamics (CFD) simulations are practical tools for describing and analyzing fluid flow problems. It is, however, often not feasible to use CFD models in control application due to their high expenses in terms of both CPU memory and time. Therefore, building accurate but fast reduced order models (ROMs) is an important step toward controlling complex systems because it enables a broader choice of control strategies.

The main idea of reduced order modeling is to find a low-rank approximation of high-fidelity data obtained experimentally or numerically. This can be achieved by determining the reduced space, spanned by a small number (order of ten) of basis functions, in which the ROM is defined. Proper orthogonal decomposition (POD) (Lumley, 1967) is a common approach to finding the reduced space. POD extracts the reduced basis by singular value decomposition (SVD) of the high-fidelity data. The ROM then results from a linear combination of the most essential basis functions. The coefficients associated with this linear combination are, in general, referred to as reduced coefficients.

Reduced order modeling techniques can be divided into two categories: *intrusive* and *non-intrusive* methods. In intrusive methods, the reduced coefficients are determined by solving ordinary differential equations obtained after projecting the full order model (FOM) onto the reduced space (see, e.g., Benner et al. (2015); see e.g. Berner et al. (2020) for a recent application to an energy engineering example). In contrast, non-intrusive methods approximate the reduced coefficients using a regression model without the need to access the governing equations of the FOM. Artificial neural networks are very good regression models.

In the intrusive context, Galerkin-projection ROMs have been developed for unsteady flows (see, e.g., Deane et al. (1991); John et al. (2010)). Some of these studies developed parameter-dependent ROMs (Amsallem and Farhat, 2008). However, these ROMs are sometimes unstable and require post-processing fitting to the original data. On that account, other projection-based techniques, such as Petrov-Galerkin, can be used to build more accurate and long-term stable ROMs (see, e.g., Reineking et al. (2022)). On the other hand, and with the significant development in machine learning techniques, neural networks have emerged as a tool to predict the temporal evolution of reduced coefficients (see, e.g., Pawar et al. (2019)). Some

⋆ Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 422037413 – TRR 287. Supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) through the AiF (German Federation of Industrial Research Associations eV) based on a decision taken by the German Bundestag (IGF no. 20275 N).

authors used a long short-term memory (LSTM) neural network in the non-intrusive framework (Maulik et al., 2021).

The present paper aims to compare the Galerkin-projection method with the non-intrusive neural network method to develop ROMs for a 2D pump model. Section 2 describes the dimensionality reduction of a dynamic system using singular value decomposition. The main ideas of projection-based and neural network-based ROMs are presented in section 3. The application example and the corresponding ROMs are presented in section 4, and a comparison is presented in section 5. A brief conclusion is stated in section 6.

2. LOW RANK APPROXIMATION OF DYNAMIC SYSTEMS

A large class of dynamical systems can be expressed in the form

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \mathbf{F}(\mathbf{q}(t)), \tag{1}$$

where $\mathbf{F}$ is a nonlinear operator, t represents the time and $\mathbf{q} \in \mathbb{R}^N$ is the state vector. The set of N ordinary differential equations (ODEs) presented in (1) are obtained, in general, from the spatial discretization of a partial differential equation with time dependence. Most discretization methods (e.g., finite element, finite difference and finite volume) result in a very high-dimensional spatial grid, i.e., $N \gg 1$, which makes the analysis and control of these systems a very challenging tasks. Therefore, we need to reduce the dimensionality of (1).

We start by solving the high dimensional model presented in (1) and we obtain at each time instant t_m, for $m = 1, \cdots, M$, the solution $\mathbf{q}(t_m) \in \mathbb{R}^N$ in the numerical domain. The snapshot matrix $\mathbf{S} \in \mathbb{R}^{N\times M}$ is obtained by concatenating all solutions at different time instants

$$\mathbf{S} = [\mathbf{q}(t_1)\ \mathbf{q}(t_2)\ \cdots\ \mathbf{q}(t_M)]\,. \tag{2}$$

The singular value decomposition (SVD) of the snapshot matrix $\mathbf{S}$ yields

$$\mathbf{S} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{V}^T, \tag{3}$$

where $\mathbf{\Phi} \in \mathbb{R}^{N\times N}$, $\mathbf{V} \in \mathbb{R}^{M\times M}$ are orthonormal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{N\times M}$ is a rectangular diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Here, $r = \mathrm{rank}(\mathbf{S})$ is the number of non-zero singular values. According to Eckart-Young theorem (Eckart and Young, 1936), the optimal low-rank approximation $\widetilde{\mathbf{S}}$ of the snapshot matrix $\mathbf{S}$, in a least-squares sense, is given by

$$\underset{\widetilde{\mathbf{S}},\ \mathrm{s.t.rank}(\widetilde{\mathbf{S}})=d}{\operatorname{argmin}} \quad \|\mathbf{S} - \widetilde{\mathbf{S}}\|_F = \widetilde{\mathbf{\Phi}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^T, \tag{4}$$

where $\widetilde{\mathbf{\Phi}} \in \mathbb{R}^{N\times d}$ and $\widetilde{\mathbf{V}} \in \mathbb{R}^{M\times d}$ are the leading d columns of $\mathbf{\Phi}$ and $\mathbf{V}$, and $\widetilde{\mathbf{\Sigma}} \in \mathbb{R}^{d\times d}$ is the leading $d \times d$ matrix of $\mathbf{\Sigma}$. $\|\cdot\|_F$ is the Frobenius norm. The columns of $\widetilde{\mathbf{\Phi}}$, denoted by $\boldsymbol{\phi}_k = [\boldsymbol{\phi}_k(q_1) \dots \boldsymbol{\phi}_k(q_N)]^T$, $k = 1, \dots, d$, are known as spatial, or POD modes, and they form an optimal basis for the reduced space $\mathbb{V}$ where the ROM lives, i.e., $\mathbb{V} = \mathrm{span}\{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_d\} \subset \mathbb{R}^N$. Therefore, the low-rank approximation of the solution vector $\widetilde{\mathbf{q}}(t_m)$ can be written as a linear combination of the basis of $\mathbb{V}$

$$\widetilde{\mathbf{q}}(t_m) = \sum_{k=1}^{d} \boldsymbol{\phi}_k a_k(t_m), \tag{5}$$

where $a_k(t_m)$ is the reduced coefficient associated to every spatial mode $\boldsymbol{\phi}_k$. The dimension d of the reduced space $\mathbb{V}$, i.e., the number of spatial modes, is determined such that

$$\epsilon = 1 - \frac{\sum_{i=1}^{d} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}, \tag{6}$$

is sufficiently small.

3. REDUCED ORDER MODELS

We divide the reduced order modeling techniques into two main categories, intrusive and non-intrusive ROMs.

3.1 Intrusive ROM using the Galerkin-projection

This method consists of two main steps. First, we insert (5) in the governing equation of the model presented in (1)

$$\sum_{k=1}^{d} \boldsymbol{\phi}_k \frac{\mathrm{d}a_k}{\mathrm{d}t} = \mathbf{F}(\sum_{k=1}^{d} \boldsymbol{\phi}_k a_k(t)). \tag{7}$$

Second, we project the resulting approximation onto the spatial modes $\boldsymbol{\phi}_k$, $k = 1, \dots, d$. We take advantage of the orthonormality of the POD modes, i.e.,

$$\langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle = \delta_{kl} = \begin{cases} 0, & \text{if } k \neq l \\ 1, & \text{if } k = l \end{cases}. \tag{8}$$

Here, $\langle \cdot, \cdot \rangle$ denote the inner products in $\mathbb{R}^N$ and δ_{kl} is the Kronecker delta. This yields a set of d ODEs

$$\frac{\mathrm{d}a_k}{\mathrm{d}t} = \langle \boldsymbol{\phi}_k, \mathbf{F}(\sum_{k=1}^{d} \boldsymbol{\phi}_k a_k(t)) \rangle \quad k = 1, \cdots, d. \tag{9}$$

Equation (9) shows that solving the system of ODEs still requires evaluation of the nonlinear operator $\mathbf{F}$ and inner products in the original dimension since $\boldsymbol{\phi}_k \in \mathbb{R}^N$. However, it is possible in many cases, e.g., Navier-Stokes equation, to compute them once in an offline step. Despite the linear structure of $\mathbb{V}$, the Galerkin-projection method is capable of generating reliable reduced models to characterize the nonlinear dynamics within $\mathbb{V}$ because it effectively captures dominant patterns of behavior by projecting onto an optimal subspace that best represents the dynamics.

3.2 Non-intrusive ROM using neural networks

The main idea of this method is to build a neural network to predict the temporal evolution of the reduced coefficients $a_k(t)$. The neural network serves as a nonlinear regression model that replaces the set of ordinary differential equations presented in (9). It is sometimes necessary to bypass the projection step, especially when the governing equations are unknown and we try to build ROM based on measurement data only.

The best reduced coefficient values associated to the reduced basis at time instant t_m will be denoted as $\mathbf{a}^{\mathrm{ref}}(t_m) \in \mathbb{R}^d$, and can be obtained by projecting the high fidelity solution $\mathbf{x}(t_m)$ onto $\mathbb{V}$, i.e.,

$$\mathbf{a}^{\mathrm{ref}}(t_m) = \langle \mathbf{q}(t_m), \widetilde{\mathbf{\Phi}} \rangle. \tag{10}$$

Therefore, using $\mathbf{a}^{\mathrm{ref}}(t_m)$, for $m = 1, \cdots, M$, as training data, we fit a neural network to predict the temporal evolution of the reduced coefficients. More specifically, we

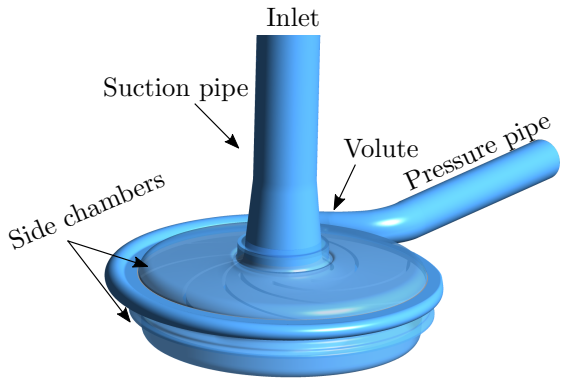

Fig. 1. 3D Model of the pump.

predict the reduced coefficients at time step t_{m+1} based on the previous h time steps,

$$\mathbf{a}(t_{m+1}) = \tilde{f}(\mathbf{a}(t_m), \cdots, \mathbf{a}(t_{m-h})), \tag{11}$$

where $\tilde{f}$ is a non-linear function and h is a hyperparameter. In this context, a long short-term memory (LSTM) neural network is a very good candidate because of its ability to learn sequence data using a gating mechanism. The reader is referred to Hochreiter and Schmidhuber (1997) for more details about LSTM layers.

4. APPLICATION TO A PUMP MODEL

We apply the two aforementioned methods to build ROMs for a hydraulic system. We consider a 3D radial pump characterized by a low specific speed ($n_s = 12\,1/\text{min}$, see Fig. 1 for a sketch). It consists of an impeller, a spiral volute, side chambers, and suction and pressure pipes. The system dynamics are modeled with the incompressible Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\Delta\mathbf{u} - \nabla p, \tag{12a}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{12b}$$

where $\mathbf{u}$, ν, and p denote the velocity vector, viscosity, and pressure, respectively. Equations (12a) and (12b) are solved using a finite volume method by discretizing the numerical domain into 1.8 million cells. This model will be denoted as the full-order model (FOM). A Dirichlet inlet boundary condition is set for velocity ($\|\mathbf{u}\| = 2.12\,\text{m/s}$) together with a zero-gradient condition for static pressure. At the outlet, zero-gradient boundary conditions are set for velocity components and Dirichlet condition for static pressure. The angular velocity of the impeller is $\omega = 151.84\,\text{s}^{-1}$ (See Sommer et al. (2023) and references therein for more details on the application).
Open source software OpenFOAM (Weller et al., 1998) is used to obtain the 3D high-fidelity solution. We aim to build a ROM for a 2D axial section of the flow field solution at the mid-span of the impeller. For this purpose, we interpolate the 3D time-variant solution onto a fixed, uniform cartesian grid, and we treat the moving impeller solid domain as a fluid by interpolating values from the surrounding blade-wall adjacent flow fields (see Fig. 4 (a)).

4.1 Intrusive ROM of the fluid flow

We denote by $\mathbf{u}(\mathbf{x}_n, t_m) = [u_x(\mathbf{x}_n, t_m) \quad u_y(\mathbf{x}_n, t_m)]^T \in \mathbb{R}^2$ as the vector of velocities at every discrete time step t_m, $m = 1, \cdots, M$, and cell with spatial coordinates $\mathbf{x}_n$, $n = 1, \cdots, N_{\text{grid}}$. First, we decompose $\mathbf{u}(\mathbf{x}_n, t_m)$ into temporal mean $\overline{\mathbf{u}}(\mathbf{x}_n)$ and fluctuation $\mathbf{u}'(\mathbf{x}_n, t_m)$ components

$$\begin{aligned}\mathbf{u}(\mathbf{x}_n, t_m) &= \overline{\mathbf{u}}(\mathbf{x}_n) + \mathbf{u}'(\mathbf{x}_n, t_m)\\ &= \frac{1}{M}\sum_{m=1}^{M}\mathbf{u}(\mathbf{x}_n, t_m) + \mathbf{u}'(\mathbf{x}_n, t_m).\end{aligned} \tag{13}$$

At each time instant t_m, we obtain the fluctuation component of the velocity vector $\mathbf{u}'(\mathbf{x}_n, t_m)$ in the numerical domain, and collect it to obtain a $2N_{\text{grid}} \times 1$ column vector of the snapshot matrix as shown in (2). For consistency with the procedure presented in section 2, we will denote $2N_{\text{grid}}$ by N. In this work, we assume that the flow is periodic, and we collect $M = 52$ equidistant snapshots. The temporal discretization involves establishing an interface between stator and rotor domains, ensuring each time step equates to a 1° rotation of impeller blades, resulting in one blade passage every 52°. Following the procedure explained in sections 2 and 3 we obtain the reference modal coefficients

$$a_k^{\text{ref}}(t_m) = \sum_{n=1}^{N_{\text{grid}}} \widetilde{\mathbf{u}}'(\mathbf{x}_n, t_m)\cdot\boldsymbol{\phi}_k(\mathbf{x}_n), \tag{14}$$

and the projection-based ROM in terms of d ordinary differential equations for the reduced coefficients $a_k^{\text{GP}}(t)$, $k = 1, \ldots, d$,

$$\frac{\mathrm{d}a_k^{\text{GP}}(t)}{\mathrm{d}t} = \sum_{i=1}^{d}\sum_{l=1}^{d} a_i(t)a_l(t)q_{kil} + \sum_{i=1}^{d} a_i(t)l_{ki} + c_k, \tag{15a}$$

where the coefficients result from the projection of (12) onto the modes $\boldsymbol{\phi}_k$

$$\begin{aligned}
q_{kil} &= -\sum_{n=1}^{N_{\text{grid}}} \boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\big(\boldsymbol{\phi}_i(\mathbf{x}_n)\cdot\nabla\big)\boldsymbol{\phi}_l(\mathbf{x}_n),\\
l_{ki} &= \sum_{n=1}^{N_{\text{grid}}} \Big(\nu\boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\Delta\boldsymbol{\phi}_i(\mathbf{x}_n)\\
&\qquad - \boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\big(\bar{u}(\mathbf{x}_n)\cdot\nabla\big)\boldsymbol{\phi}_i(\mathbf{x}_n)\\
&\qquad - \boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\big(\boldsymbol{\phi}_i(\mathbf{x}_n)\cdot\nabla\big)\bar{u}(\mathbf{x}_n)\Big),\\
c_k &= \sum_{n=1}^{N_{\text{grid}}} \Big(-\boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\big(\bar{u}(\mathbf{x}_n)\cdot\nabla\big)\bar{u}(\mathbf{x}_n)\\
&\qquad + \nu\boldsymbol{\phi}_k(\mathbf{x}_n)\cdot\Delta\bar{u}(\mathbf{x}_n)\Big),
\end{aligned} \tag{15b}$$

with $i, l = 1, \ldots, d$. The reader is referred to John et al. (2010) for more details about the derivation of (15). To ensure that the reduced coefficients $a_k^{\text{GP}}(t_m)$ fit the original data as good as possible, we perform an optimization of the ROM coefficients q_{kil}, l_{ki} and c_k from (15b) (see, e.g., Sommer et al. (2023); Couplet et al. (2005)). For this purpose, we solve

$$\min_{q_{kil}, l_{ki}, c_k} \sum_{k=1}^{d}\sum_{m=1}^{M} (a_k^{\text{GP}}(t_m) - a_k^{\text{ref}}(t_m))^2, \tag{16}$$

for $k = 1, \ldots, d$ and $m = 1, \ldots, M$. We use the resulting coefficients and denote the solution of (15a) with these coefficients by $a_k^{\text{GP,fit}}$.
The approximated velocity vector with this solution at some time instant t_m and spatial point $\mathbf{x}_n$ is

$$\mathbf{u}^{\text{GP,fit}}(\mathbf{x}_n, t_m) = \overline{\mathbf{u}}(\mathbf{x}_n) + \sum_{k=1}^{d} \boldsymbol{\phi}_k(\mathbf{x}_n)a_k^{\text{GP,fit}}(t_m). \tag{17}$$

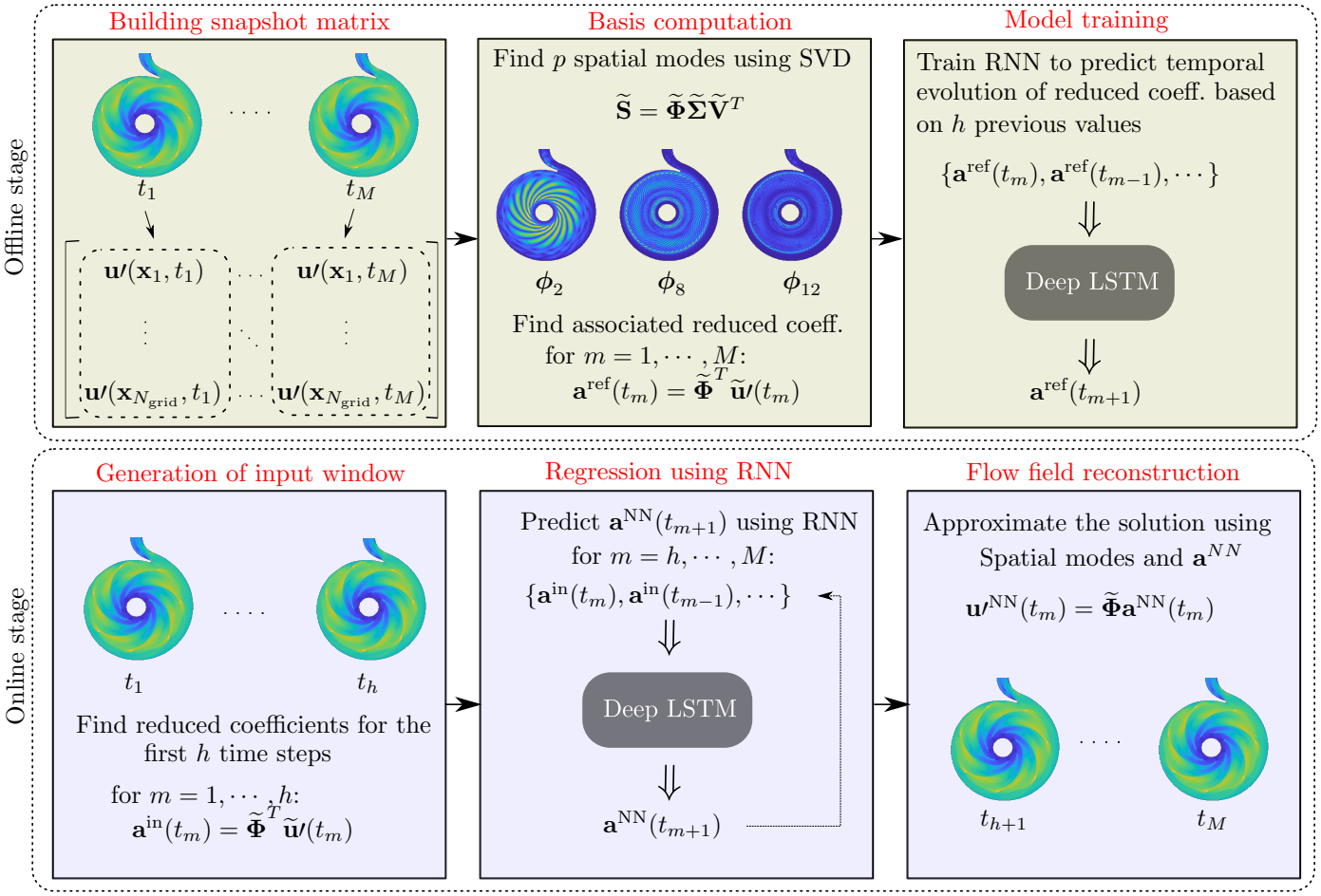

Fig. 2. Data-driven ROM framework.

Table 1. Hyperparameters of the neural network

Hyperparameter	
number of previous steps h	3
number of layers	4
number of hidden states per layer	89
learning rate	0.001
batch size	8
epochs	500
loss function	Mean squared error
optimizer	Adam

4.2 Non-intrusive ROM of the fluid flow

The steps to build the non-intrusive ROM are summarized in Fig. 2. The first two steps in the offline stage, namely building the snapshot matrix and finding the reduced basis, are common with the Galerkin-projection method explained above. In the third step, we build and train a deep recurrent neural network with LSTM layers to predict the reduced coefficients. The hyperparameters of the neural network are presented in Tab. 1. It should be noted that in this work, we tuned the hyperparameters manually to get the best prediction performance. However, other methods such as Bayesian optimization (Brochu et al., 2010) and random search (Bergstra and Bengio, 2012) can be used to find the optimal set of hyperparameters.

In order to accelerate training, we normalize the reduced coefficients between 0 and 1. We retain 20% of the training samples for validation purposes. We use the open-source library TensorFlow to build and train the neural network (Abadi, 2016). Once the model is trained, it can be used to predict the reduced coefficients at any time instant t_m. An input window with h initial conditions must be generated for the first prediction, as shown in the online stage in Fig. 2. We generate the input window based on the CFD results at the first h time steps. We denote by $\mathbf{a}^{\text{NN}}(t_{m+1})$ as the vector of reduced coefficients associated with all spatial modes at time instant t_{m+1} predicted by the neural network. The prediction of the reduced coefficients in this method is performed in a recursive manner, i.e., the outputs of the neural network are used as inputs in the next prediction.

The final step in the online stage consists of reconstructing the velocity vector in a similar way to (17), but using the neural network predictions $a_k^{\text{NN}}(t_m)$ of the reduced coefficients, i.e.,

$$\mathbf{u}^{\text{NN}}(\mathbf{x}_n, t_m) = \overline{\mathbf{u}}(\mathbf{x}_n) + \sum_{k=1}^{d} \phi_k(\mathbf{x}_n) a_k^{\text{NN}}(t_m). \qquad (18)$$

5. RESULTS

A comparison between Galerkin-projection ROM, referred to as ROM-GP, and the non-intrusive neural network ROM, referred to as ROM-NN, is presented in this section. The comparison is performed with respect to the following four aspects.

Table 2. Comparison of the computational time

	ROM-GP	ROM-NN	FOM
Offline-time (s)	4145.663[1]	53.265[1]	-
Online-time (s)	0.015[1]	1.102[1]	3061[2]

[1] Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz
[2] Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz

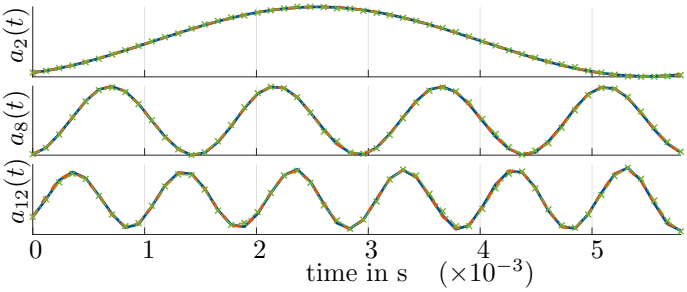

Fig. 3. Coefficients a_k^{ref} from (14) (blue, solid line), $a_k^{\text{GP,fit}}$ from (15a) (red, dashed line) and a_k^{NN} from (11) (green crossed) for a single period (corresponding to $M = 52$), with $k = 2, 8, 12$.

5.1 Computational time

To begin, we compare the computational offline-time required to build each ROM. Both ROMs were built and evaluated on the same hardware. The results are presented in Tab. 2. It can be seen that the offline-time of ROM-GP is two orders of magnitude greater than the offline-time of ROM-NN. The main reason for this is that most of the offline-time of ROM-GP is consumed by the optimization step presented in (16) which fits the reduced coefficients $\mathbf{a}^{\text{GP}}(t)$ to the reference values $\mathbf{a}^{\text{ref}}(t)$. The offline time for creating the ROM-NN is consumed mainly by the training step. We stress that, while training of NN is indeed time-consuming, it requires less computation time than generating the ROM-GP.
Once the ROM is optimized respectively trained, it can be used for an online evaluation of the flow field solution at instant t_m. Table 2 shows that the online evaluation is much faster for both ROMs than for the FOM, even if the FOM is evaluated on a much more computationally powerful hardware.

5.2 Relative error

We compare the predicted reduced coefficients $\mathbf{a}^{\text{GP,fit}}(t)$ and $\mathbf{a}^{\text{NN}}(t)$ with the reference $\mathbf{a}^{\text{ref}}(t)$. The results are plotted in Fig. 3 for the reduced coefficients corresponding to modes 2,8 and 12. It can be seen that both $\mathbf{a}^{\text{GP,fit}}(t)$ and $\mathbf{a}^{\text{NN}}(t)$ are in very good agreement with $\mathbf{a}^{\text{ref}}(t)$. We further evaluate the resulting velocity ROMs by reconstructing the flow field solution at some instant t_m using (17) and (18) for ROM-GP and ROM-NN, respectively. We report the normalized and averaged velocity error as

$$\varepsilon = \frac{1}{2N_{\text{grid}}M} \sum_{n=1}^{N_{\text{grid}}} \sum_{m=1}^{M} \frac{\|\mathbf{u}(\mathbf{x_n}, t_m) - \mathbf{u}^{\text{ROM}}(\mathbf{x_n}, t_m)\|}{u_{\text{ref}}}, \tag{19}$$

where $\mathbf{u}^{\text{ROM}}$ is the reconstructed solution and $u_{\text{ref}} = 16.7\text{m/s}$ is the rotational velocity at the outer radius of the impeller. Results show that ROM-GP, with $\varepsilon_{\text{ROM-GP}} = 0.14\%$ is slightly better than ROM-NN, with $\varepsilon_{\text{ROM-NN}} = 0.18\%$, over the first period. A detailed description for the relative error at each $\mathbf{x}_n$, evaluated at $t = 10^{-3}$ s is presented in Fig. 4. Both ROM-GP and ROM-NN are in very good agreement with the CFD solution, where the maximum local error, which occurs at the tip of the blades in all cases, never exceeds 5% for either ROM.

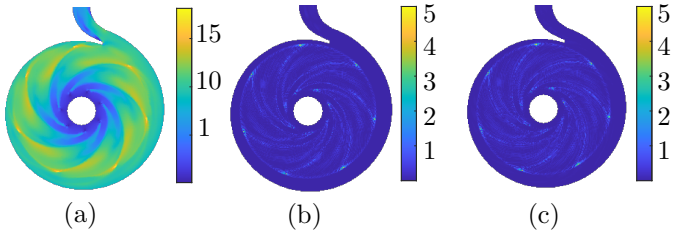

Fig. 4. (a) Instantaneous velocity field magnitude (in m/s) obtained from the interpolated CFD simulation (b) relative error (in %) between CFD and ROM-GP (c) relative error (in %) between CFD and ROM-NN.

5.3 Long-term stability

We check the stability of both methods by predicting the reduced coefficients $\mathbf{a}^{\text{GP,fit}}(t)$ and $\mathbf{a}^{\text{NN}}(t)$ for many periods T. We emphasize that both ROM-GP and ROM-NN are built from flow field data corresponding only to 1 period. Figure 5 shows that ROM-GP is not stable and fails to predict the reduced coefficients outside the optimization period. On the other hand, ROM-NN converges to a limit cycle that is close to the one formed by the reference $\mathbf{a}^{\text{ref}}(t)$. Our numerical experiments showed that, even for 200 Periods, ROM-NN remains stable.

5.4 Sensitivity to initial condition

Finally, we examine the sensitivity to initial condition for each ROM by predicting the reduced coefficients $\mathbf{a}^{\text{GP,fit}}(t)$ and $\mathbf{a}^{\text{NN}}(t)$ based on noisy measurements as initial conditions, i.e.,

$$a_k^n(t) = a_k^{\text{ref}}(t) + N(0,1)\widetilde{\mathbf{\Sigma}}_{kk}, \text{ for } k = 1, \cdots, d, \tag{20}$$

where $a_k^n(t)$ is the noisy reference signal for the reduced coefficients corresponding to each spatial mode, $N(0,1)$ represents a normal distribution with 0 mean and 1 variance, and $\widetilde{\mathbf{\Sigma}}_{kk}$ is the singular value of mode k. It should be noted that the initial set corresponding to ROM-GP consists only of the reduced coefficient values at $t = 0$. However, we need h initial conditions, corresponding to $t = 0, \cdots, h$, for ROM-NN. The results are plotted in Fig. 6. It is evident that ROM-GP does not tolerate perturbation in the initial condition, especially for the reduced coefficients corresponding to higher spatial modes. In contrast, ROM-NN is more robust despite requiring more initial conditions in the input window.

The results indicate that ROM-NN outperforms ROM-GP slightly. This is because the dynamical model of ROM-GP presented in (15) is built using only dominant modes, neglecting less dominant ones. The improvement of ROM-GP requires inclusion of closure terms to compensate for this loss of information (see, e.g., Ahmed et al. (2021)). In contrast, ROM-NN has no restrictions on the dynamical model, allowing it to find the best-fitting function for the reduced coefficients in the reduced space.

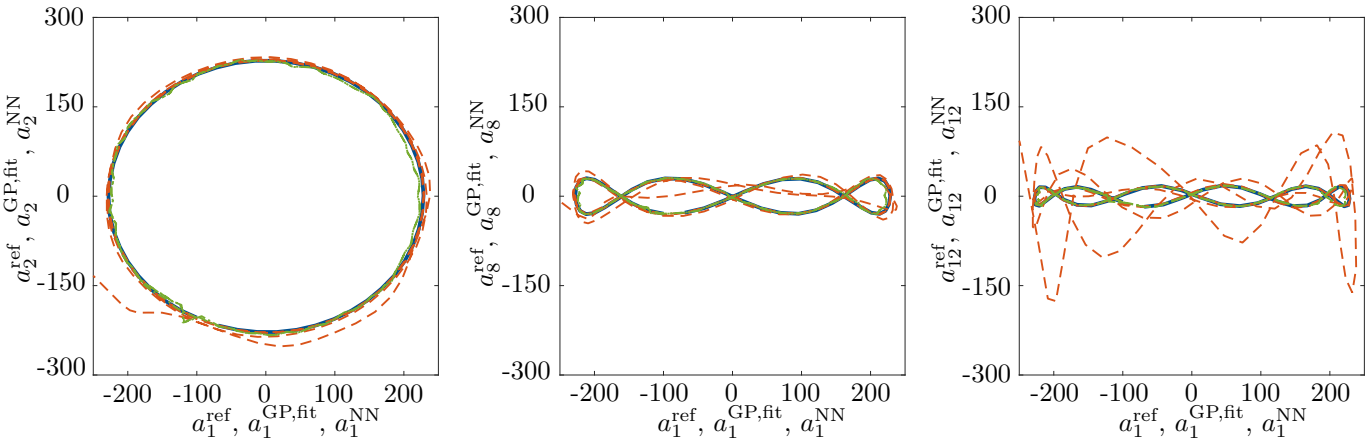

Fig. 5. Orbits of reduced coefficients: a_k^{ref} marked with blue solid line, $a_k^{\mathrm{GP,fit}}$ for 4 periods marked with red dashed line, and a_k^{NN} for 4 periods marked with green dash-dotted line.

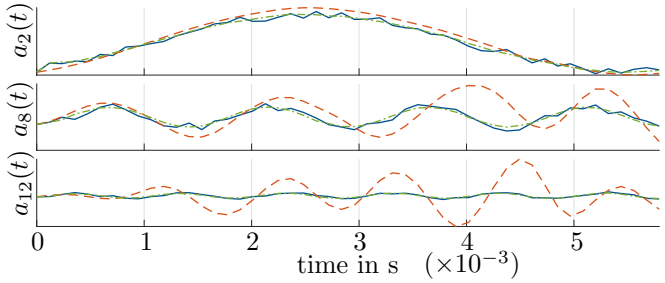

Fig. 6. Noisy a_k^{n} signal used to find the initial condition set (blue, solid line), $a_k^{\mathrm{GP,fit}}$ from (15a) (red, dashed line) and a_k^{NN} from (11) (green, dash-dotted line) for a single period (corresponding to $M = 52$), with $k = 2, 8, 12$.

6. CONCLUSION

Real-time control of high-dimensional dynamic models is still impractical with the available computational resources. Therefore, building ROMs is a necessary step. In this work, we compared standard Galerkin-projection with a non-intrusive neural network method to build reduced-order models for a 2D centrifugal pump model. While the developed ROMs correspond to the pump model with a fixed impeller angular velocity, enabling control of the considered pump model involves controlling the rotation speed of the impeller to achieve the desired velocity or pressure fields. In the future, we aim to address this aspect by extending our ROMs to account for variations in input parameters.

REFERENCES

Abadi, M. (2016). TensorFlow: A system for Large-Scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283. USENIX Association, Savannah, GA.

Ahmed, S.E., Pawar, S., San, O., Rasheed, A., Iliescu, T., and Noack, B.R. (2021). On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. *Physics of Fluids*, 33(9), 091301.

Amsallem, D. and Farhat, C. (2008). Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 47(7), 1803–1813.

Benner, P., Gugercin, S., and Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4), 483–531.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281–305.

Berner, M.O., Scherer, V., and Mönnigmann, M. (2020). Controllability analysis and optimal control of biomass drying with reduced order models. *Journal of Process Control*, 89, 1–10.

Brochu, E., Cora, V.M., and de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. misc.

Couplet, M., Basdevant, C., and Sagaut, P. (2005). Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207, 192–220.

Deane, A., Kevrekidis, I.G., Karniadakis, G., and Orszag, S.A. (1991). Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Physics of Fluids*, 3, 2337–2354.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

John, T., Guay, M., Hariharan, N., and Naranayan, S. (2010). POD-based observer for estimation in Navier-Stokes flow. *Computers and Chemical Engineering*, 34(6), 965 – 975.

Lumley (1967). The structure of inhomogeneous turbulence. *Atmospheric Turbulence and Wave Propagation*, 166 – 178.

Maulik, R., Lusch, B., and Balaprakash, P. (2021). Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3), 037106.

Pawar, S., Rahman, S.M., Vaddireddy, H., San, O., Rasheed, A., and Vedula, P. (2019). A deep learning enabler for non-intrusive reduced order modeling of fluid flows. *Physics of Fluids*, 31(8), 085101.

Reineking, L., Sommer, K.D., Ravichandran, Y.P., Skoda, R., and Mönnigmann, M. (2022). Long-term stable reduced models for hydraulic systems governed by reynolds averaged navier-stokes equations. *IFAC-PapersOnLine*, 55(7), 254–259.

Sommer, K.D., Reineking, L., Ravichandran, Y.P., Skoda, R., and Mönnigmann, M. (2023). Estimating flow fields with reduced order models. *Heliyon*, e20930.

Weller, H.G., Tabor, G., Jasak, H., and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6), 620–631.