biomimetics

MDPI

Article

Deep-Learning-Based Reduced-Order Model for Power Generation Capacity of Flapping Foils

Ahmad Saeed [1], Hamayun Farooq [1,2], Imran Akhtar [1], Muhammad Awais Tariq [2] and Muhammad Saif Ullah Khalid [3,*]

[1] Department of Mechanical Engineering, NUST College of Electrical & Mechanical Engineering, National University of Sciences & Technology, Islamabad 46000, Pakistan
[2] Department of Mathematics and Statistics, Institute of Southern Punjab (ISP), Multan 60800, Pakistan
[3] Department of Mechanical Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada
[*] Correspondence: mkhalid7@lakeheadu.ca

Abstract: Inspired by nature, oscillating foils offer viable options as alternate energy resources to harness energy from wind and water. Here, we propose a proper orthogonal decomposition (POD)-based reduced-order model (ROM) of power generation by flapping airfoils in conjunction with deep neural networks. Numerical simulations are performed for incompressible flow past a flapping NACA-0012 airfoil at a Reynolds number of 1100 using the Arbitrary Lagrangian–Eulerian approach. The snapshots of the pressure field around the flapping foil are then utilized to construct the pressure POD modes of each case, which serve as the reduced basis to span the solution space. The novelty of the current research relates to the identification, development, and employment of long-short-term neural network (LSTM) models to predict temporal coefficients of the pressure modes. These coefficients, in turn, are used to reconstruct hydrodynamic forces and moment, leading to computations of power. The proposed model takes the known temporal coefficients as inputs and predicts the future temporal coefficients followed by previously estimated temporal coefficients, very similar to traditional ROM. Through the new trained model, we can predict the temporal coefficients for a long time duration that can be far beyond the training time intervals more accurately. It may not be attained by traditional ROMs that lead to erroneous results. Consequently, the flow physics including the forces and moment exerted by fluids can be reconstructed accurately using POD modes as the basis set.

Keywords: power generation; long-short-term neural network; proper orthogonal decomposition; flapping foils; reduced-order modeling

check for updates

Citation: Saeed, A.; Farooq, H.; Akhtar, I.; Tariq, M.A.; Khalid, M.S.U. Deep-Learning-Based Reduced-Order Model for Power Generation Capacity of Flapping Foils. *Biomimetics* **2023**, *8*, 237. https://doi.org/10.3390/biomimetics8020237

Academic Editors: Isa Ebtehaj and Sayed M. Bateni

Received: 23 February 2023
Revised: 26 May 2023
Accepted: 2 June 2023
Published: 5 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Despite enormous advancements in computer-related technologies in the modern era, performing numerical simulations of complex flows using computational fluid dynamics (CFD) based tools demands a lot of resources in terms of computing time and data storage capacity. To cope up with these challenges, the idea of developing reduced-order models (ROMs), capturing the dynamics of engineering systems, is vital. The essence of such ROMs is to simulate the behavior of a system for a chosen set of values for the control parameters in the governing mathematical equations. For systems involved with fluid flows, the actual models usually consist of a system of nonlinear partial differential equations (PDEs). When the solutions are computed for a specific set of parameters, we can build *basis functions*. These functions help generate approximate solutions for new values of the governing parameters inexpensively. With the advantages offered by different classes of ROMs, such models were extensively developed for numerous applications, involving optimizations and control of complex engineering systems [1–5]. The success of a ROM highly depends on the choice of *basis functions* to be used to approximate the solution. Although a variety of dimensionality-reduction techniques exist, most ROMs are based on proper orthogonal decomposition (POD) technique [6,7] serving as the *basis functions*. The POD method is very effective for dimensionality reduction complex models of physical systems. Snapshots (measurements) of many nonlinear dynamical systems often exhibit low-dimensional activity, and their contribution in a whole phenomenon is minimal. It is because most of the energy (quantified through the variance of a signal) is contained in a few modes computed through singular value decomposition (SVD) or method of snapshots [8]. Such modes serve as the *reduced-order basis* to construct the low-dimensional space for the construction of a ROM. The significance of the POD method in the development of ROMs can be found in numerous fields, including complex flows [9], flow control design [1,10–13], pattern recognition [8], and uncertainty quantification and optimization [14]. The application of such POD-ROMs in turbulent flows is not straightforward. Some recent pieces of work on closure modeling attempts to extend the application of POD-ROMs in complex flows [9,15–18]. Generally, the construction of a POD-ROM is a two-step process: (a) post-processing of flow field data to compute the optimal basis functions, and (b) Galerkin projection of the governing equations onto these basis functions to develop a ROM. Linear combinations of POD modes and expansion coefficients (temporal coefficients) are employed to represent a time-evolving flow field. POD-Galerkin-based models suffer from issues related to instability and efficiency [18–23].

Although a significant reduction in computational time and cost may be attained to find the solution of complex phenomena by using ROMs, there exists a lot of room for researchers to formulate better alternates for further improvements. The inclusion of machine learning (ML) in CFD brings a revolution in this field [24–27]. Han et al. [28] introduced a deep-learning-based approach for solving highly complex systems of PDEs. They used neural networks to predict the values of unknown parameters and concluded that the solver performed well in terms of accuracy and computational cost. Hesthaven and Ubbiali [29] utilized ANN to approximate the coefficients of a reduced-order model for parameterized steady-state PDEs. Their results confirmed the speedup and accuracy of the ANN-based model.

Most ROMs based on the POD technique employ information about velocity fields [5,21] whereas, the pressure field, specifically on the surface, also plays an important role in the accuracy of POD-based ROMs [30]. Many studies were carried out for ROMs developed based on pressure mode decomposition (PMD) method that also enabled accurate estimations of hydrodynamic forces. Recently, Ahmed et al. [31] used PMD technique to develop an ML-based ROM to predict the lift and drag forces for flows around circular cylinders. Later, Farooq et al. [32] developed an ANN-based ROM to approximate the hydrodynamic forces on a NACA-0012 airfoil exerted by the fluid flows, passing over it at different angles-of-attack (AoA). The trained model produced accurate values of lift and drag forces that match well with the true values obtained through direct numerical simulations. Despite vast research in this field, the current situation demands more novel ideas and techniques stemming from ANN-based ROMs for accurate prediction of hydrodynamic forces on flapping foils similar to the conventional ROMs, where the solution can be predicted for a long time duration.

Most of the research studies of flapping foils/wings [33–37] are focused more to examining biological and bio-inspired propulsion mechanisms with the primary objective of developing efficient engineered propulsive devices. Alternatively, the flapping foils/wings can also be employed to harness energy from the fluid flows, passing over them. Flapping foils-based power extraction systems are usually classified into three categories according to their activation mode [38,39]: (i) a fully forced system in which both plunge and pitch motions are prescribed [38,40–44]; (ii) a semi-passive system in which the pitch motion is prescribed while the plunge motion is induced from the interactions between the foil, the incoming flow, and the elastic supports [45–50]; and (iii) a fully-passive system in which both plunge and pitch motions are entirely driven by interactions of the foil with the surrounding fluid and the supporting dynamical mechanisms [42,44,51–54]. In this work, we consider a fully forced system for developing a deep-learning-based ROM for power generation by flapping foils. Generally, the power extraction performance of a flapping foil is measured through a parameter called power extraction efficiency [41]. Kinsey and Dumas [41] conducted a numerical parametric study to investigate the performance of the fully forced system of a NACA-0015 airfoil at a Reynolds number (Re) of 1100 for the range of flapping frequency and pitching amplitude. The main objective of their study was to determine the ranges for the optimal frequency and pitching amplitude for maximum power extraction efficiency. Their study demonstrated that the power extraction efficiency could exceed 20% when setting the pitching amplitude greater than 55°. It reaches its maximum value of 34% when considering the plunging amplitude equal to the foil's chord length and selecting the pitching amplitude higher than 75°. In another numerical study conducted by Ashraf [55], the power extraction efficiency of NACA-0014 airfoil at Re = 20,000 was examined. He investigated the effect of the phase angle between the plunge and pitch kinematics. The peak value of the power coefficient and efficiency (32%) was achieved at the phase angle of 95° with a plunging amplitude equal to 5% more than the foil's chord length. Recently, Farooq et al. [40] conducted a numerical study to investigate the power extraction performance of a fully forced NACA-0012 airfoil at Re $= 1100$. They performed a parametric study by varying the Strouhal number and the amplitude of the pitching angle to identify two operational flow regimes: power generation and thrust-producing propulsion using the feathering criterion [41]. Their parametric study revealed that the foil could reach up to 42% power generation efficiency when setting the pitching amplitude in the range of 60° to 70° with Strouhal number synchronized with the non-dimensional excitation frequency. Moreover, they identified the locations where the fluid pressure was dominant during the oscillating cycle of the flapping airfoil, operating in the power generation regime and designed a piezoelectric energy harvester for electrical power production [56].

The novelty of our present research is to develop and employ a recurrent neural network (RNN) [57] model to predict temporal coefficients of the pressure modes. These coefficients, in turn, are used to reconstruct hydrodynamic forces and moment, leading to computations of power. It is important to mention here that, in a conventional ROM, temporal coefficients are evaluated by integrating the time-evolving dynamical system, which is usually a set of initial-valued ordinary differential equations and can lead to erroneous results for complex flows. However, in ML-based ROMs, these temporal coefficients can be predicted accurately by employing an RNN-based model [58,59]. Note that the pressure temporal coefficients are not only required for the reconstruction of hydrodynamic forces and moment but also are useful for the development of efficient ROMs for shear flows [30]. The main focus of the current study is to develop a ROM for a power generation system using a variant long-short-term neural network (LSTM) for the prediction of temporal data. Our newly proposed model is very similar to a traditional ROM, which takes the known temporal coefficients as input and predicts the future temporal coefficients followed by previously estimated temporal coefficients. Consequently, we can predict the temporal coefficients for a long time duration that can be far beyond the training time intervals more accurately, which may not be attained by traditional ROMs that lead to erroneous results. The model's efficiency is verified by reconstructing hydrodynamic forces and moment accurately. Note that for reconstruction, we do not employ the training temporal data. The motivation behind the choice of the LSTM model compared to others is that LSTM models are particularly well-suited for tasks that require modeling sequential data with long-term dependencies. These models were previously shown to perform well in a wide range of applications for reduced-order modeling of fluid flow problems [60–63], which made them a popular choice in many deep-learning tasks. LSTM models have some advantages over other models, such as for capturing long-term dependencies. These techniques can overcome the “vanishing gradient" problem, as explained in Section 5.3. Handling variable input sequence length, they do not need fixed inputs and output sizes, unlike *feedforward* neural networks. Having memory and recurrent connections, they models can store and propagate information over time steps, which is useful for tasks, requiring temporal dependencies.

The remaining manuscript is organized as follows. The governing mathematical models for incompressible two-dimensional flows over flapping foils are presented in Section 2. In addition, important details about our computational techniques are presented here. Next, the validation studies of our numerical methodology is provided in the Section 4. Details on the feature extraction technique, POD of the pressure fields, using the method of snapshots and the ANN training model and its behavior constitute Section 5.

2. Numerical Methodology

2.1. Governing Formulations for Incompressible Fluid Flows

The mathematical model, describing the two-dimensional (2D) unsteady incompressible flows around flapping airfoils consists of the continuity and Navier-Stokes equations, which are defined in their respective tensor as given below:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u_i}{\partial x_j^2}\right) \tag{2}$$

where the u_i represents the fluid velocity, p is the pressure, and ρ is the density of the fluid. This governing model (Equations (1) and (2)) is provided in the nondimensional form, where nondimensionalization is carried out by considering the free-stream uniform velocity (U_∞) as the velocity scale and the chord-length (c) of the airfoil as the length scale. Thus, the Reynolds number (Re) is defined as $\mathrm{Re} = \rho c U_\infty / \mu$, where μ is dynamic viscosity of the fluid.

In order to handle the effects of moving bodies in a flowing fluid, we employ *Arbitrary Lagrangian–Eulerian* (ALE) approach in which the momentum equation (Equation (2)) is modified as:

$$\frac{\partial u_i}{\partial t} + \widetilde{u}_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u_i}{\partial x_j^2}\right) \tag{3}$$

Here, the term $\widetilde{u}_j$ represents the relative velocity of the fluid with respect to the corresponding grid node-velocity, that is, $\widetilde{u}_j = u_j - u_{g_j}$. The term u_{g_j} represents the grid node velocity, which has to be computed at each time level. Note that Equations (2) and (3) will becomes identical for $u_{g_j} = 0$, which shows *Eulerian* description and the underlying grid thus remains fixed. On the other hand, for $u_j - u_{g_j} = 0$, Equation (3) then represents the *Lagrangian* description of flow dynamics. Typically, the success of the ALE approach depends on the strategy for how the computational grid deforms during simulations. In this study, we employ the *radial basis function* (RBF) interpolation technique, for grid deformations. This method was originally proposed and developed by [64], and it is known for its robustness, accuracy, and its capability to maintain good grid quality even for large structural translations and rotations [65]. Moreover, the computation grid used here is an 'O'-type body-fitted grid around a NACA-0012 airfoil, as shown in Figure 1. Note that we consider a grid of size 400 × 304 with 400 points in the radial direction and 304 points on the surface of the airfoil. The outer radius of the circular domain around the airfoil is 30c. Moreover, the same grid size and resolution are used to compute the POD modes.

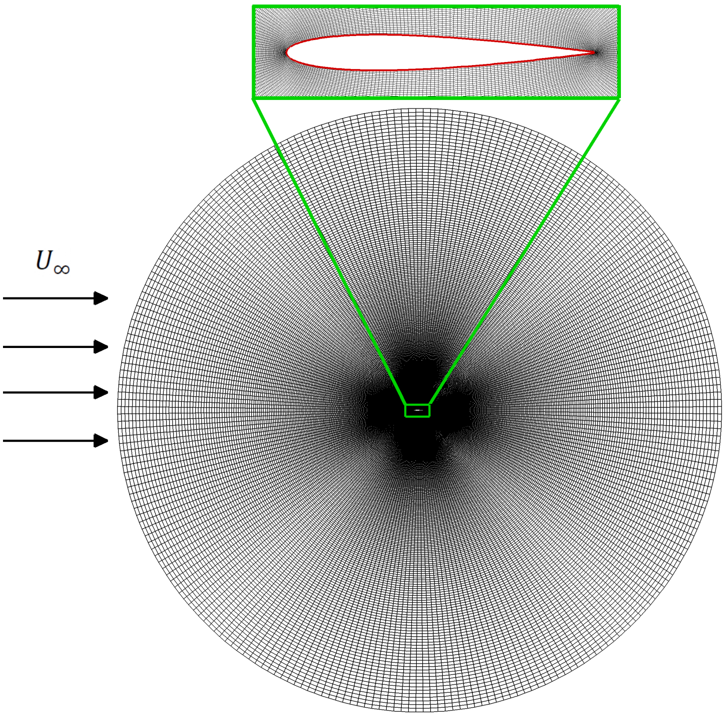


Figure 1. Two-dimensional layout of an 'O'-type body-fitted grid over the NACA0012 airfoil. Here, the horizontal arrows show the direction of the incoming flow.

Because the computational domain has curved grid lines, the following expressions are used to transform the governing equations into the curvilinear coordinates $(\alpha,\ \beta)$, depending on the Cartesian coordinates $(x,\ y)$:

$$\alpha = \alpha(x, y), \qquad \beta = \beta(x, y) \tag{4}$$

Thus, after some mathematical manipulations and simplifications, we obtain the following form:

$$\frac{\partial U_m}{\partial \alpha_m} = 0, \tag{5}$$

$$\frac{\partial (G^{-1} u_i)}{\partial t} + \frac{\partial F_{im}}{\partial \alpha_m} = 0, \tag{6}$$

where the flux is defined as

$$F_{im} = \widetilde{U}_m u_i + G^{-1} \frac{\partial \alpha_m}{\partial x_i} p - \frac{1}{\mathrm{Re}} Q^{mn} \frac{\partial u_i}{\partial \alpha_n}. \tag{7}$$

where $G^{-1} = \det\left(\frac{\partial x_i}{\partial \xi_j}\right)$ is the inverse of the Jacobian or the volume of the cell; $\widetilde{U}_m = G^{-1} \frac{\partial \alpha_m}{\partial x_j} \widetilde{u}_j$ is the volume flux (contravariant velocity multiplied by G^{-1}) normal to the surface of constant ξ_m; and $Q^{mn} = G^{-1} \frac{\partial \alpha_m}{\partial x_j} \frac{\partial \alpha_n}{\partial x_j}$ is the "mesh skewness tensor".

2.2. Prescribed Flapping Kinematics

The mathematical models for combined plunging and pitching (or flapping) kinematics are defined as:

$$y(t) = A_y \sin(2\pi f_e t), \tag{8a}$$
$$\dot{y}(t) = A_y(2\pi f_e)\cos(2\pi f_e t) \tag{8b}$$
$$\alpha(t) = A_\alpha \sin(2\pi f_e t + \frac{\pi}{2}), \tag{8c}$$
$$\dot{\alpha}(t) = A_\alpha(2\pi f_e)\cos(2\pi f_e t + \frac{\pi}{2}) \tag{8d}$$

where A_y and A_α are the nondimensional plunging and pitching amplitudes, respectively and f_e is the oscillating excitation frequency. The terms $y(t)$ and $\alpha(t)$ represent the instantaneous plunging and pitching displacements of the airfoil. For these kinematic settings, the Strouhal number (St_A) here is defined as the ratio between the product of the oscillating frequency (f_e) and peak-to-peak amplitude $(2A_y)$ of the trailing edge of the airfoil and the uniform velocity (U_∞), i.e., St_A=$2f_e A_y/U_\infty$. In our present study, we keep the value of f_e fixed at 0.375 Hz. Additionally, it is important to mention, the superscript $(\cdot)$ in Equation (8b),(8d) represents the time derivative.

2.3. Discretization Strategy

The fractional step method is an effective and strong candidate technique for dealing with incompressible Navier-Stokes equation. It transforms the momentum equation into the convection-diffusion equation and pressure-Poisson equation [32,40,51,66]. We utilize a non-staggered grid in which the velocity and pressure fields are computed at the center of each cell. The fluxes $(\widetilde{U}, \widetilde{V})$ are calculated on the corresponding faces at their midpoints. Except for the convective term, all the spatial derivatives are approximated using the second-order central difference method. The use of a central difference scheme for convective terms may result in unnecessary fluctuation and thus, may lead to incorrect or erroneous solutions. However, such issues can be resolved by implementing an appropriate higher-order difference scheme. For this purpose, an upwind scheme with quadratic upwinding interpolation for convective kinematics (QUICK) [67] is employed here to discretize the convective terms. To advance the solution in time, a hybrid method composed of explicit and semi-implicit schemes is used. The Crank–Nicolson (C-N) scheme is employed to discretize the diagonal viscous terms only, whereas all the remaining terms are discretized using the second-order Adams–Bash (AB-2) scheme. It is important to mention here that both these schemes are second-order accurate.

3. Hydrodynamic Performance Metrics

The hydrodynamic forces and moments on an oscillating body are produced due to pressure distribution and shear stress distribution over the surface of the body. The net effect of these terms integrated over the complete body surface gives us the resultant hydrodynamic force (R) and moment on the body, where R can be split into components; normal and axial forces, as illustrated below.

$$F_N = \oint_c dN = -\oint_c (p\sin(\theta) - \tau\cos(\theta))\,ds, \tag{9a}$$

$$F_A = \oint_c dA = -\oint_c (p\cos(\theta) + \tau\sin(\theta))\,ds, \tag{9b}$$

$$M = -\oint_c (x^b - x_p)dF_N + \oint_c (y^b - y_p)dF_A, \tag{9c}$$

where F_N and F_A are the axial and normal force components that are parallel and perpendicular to the chord of the airfoil, respectively. The term τ represents the shear stress, whereas p is the pressure. Furthermore, M is the pitching moment about the pitching axis point (x_p, y_p), and s is the arc-length of the body. Thus, the hydrodynamic lift (L) and drag (D) forces over the airfoil at the pitching angle α can be computed as:

$$L = F_N \cos(\alpha) - F_A \sin(\alpha), \tag{10a}$$

$$D = F_N \sin(\alpha) + F_N \cos(\alpha) \tag{10b}$$

The dimensionless forces and moment coefficients can be defined by using dynamic pressure ($q_\infty = \frac{1}{2}\rho U_\infty^2$) as follows:

$$C_L = \frac{L}{q_\infty c}, \tag{11a}$$

$$C_D = \frac{D}{q_\infty c}, \tag{11b}$$

$$C_M = \frac{M}{q_\infty c^2}. \tag{11c}$$

4. Validation

In order to demonstrate the effectiveness and accuracy of our in-house computational solver, we simulate the flows over the NACA-0015 airfoil, undergoing simultaneous plunging and pitching motions at Re $=$ 1100. We compute the hydrodynamic lift and drag coefficients and compare the temporal profile of C_D with those reported by [37]. Here, kinematic amplitudes are $A_h = 0.4$, $A_\alpha = 20°$, for plunging and pitching motions, respectively. Figure 2 presents that the results from our computational solver are in excellent agreement with those from [37]. A body-fitted nearly orthogonal grid of size 353 × 501 around the foil is used for our simulations. The outer boundary of this grid is located at a radius of 25c that is far from the surface of the foil, as shown in Figure 1). For simulating the flow over the moving body here, we use the time-step equal to 2×10^{-4}. For more details on grid convergence and time-step independence, interested readers are referred to the Refs. [40,51].

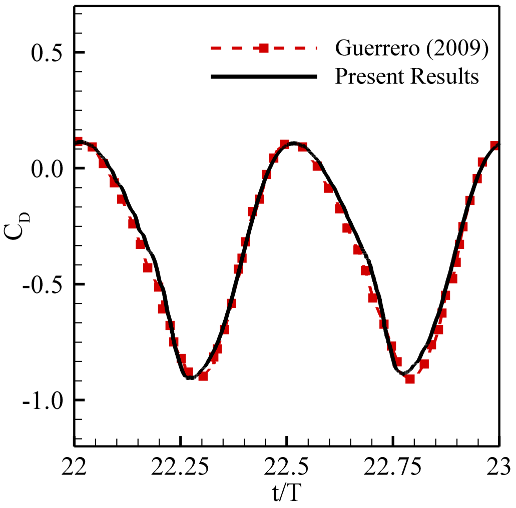


Figure 2. Validation of solver: Comparison of C_D of the flapping airfoil with the temporal data of [37]. Here T represents the oscillation time period.

5. Results and Discussions

In this study, we perform numerical simulations for incompressible flows over a NACA-0012 foil that is subjected to sinusoidal plunging and pitching (flapping) motions simultaneously at Re $= 1100$. Before analyzing the accuracy of a deep-learning neural network-based ROM for the pressure field, we first discuss the performance of a flapping foil, operating in the power generation regime. Under these conditions, the directions of the lift force and plunging velocity are mostly the same in an oscillation cycle. Thus, the flapping foil experiences a positive work by the fluid, passing over it. Such positive work can be utilized for power extraction [40,41]. We simulate several cases from the parametric space of $(\mathrm{St}_A, A_\alpha) = \{0.1 \leq \mathrm{St}_A \leq 0.35 \text{ and } 15^\circ \leq A_\alpha \leq 100^\circ\}$ and compute the power generation efficiency (η) defined as:

$$\eta = \bar{C}_P \frac{c}{Y_P} \tag{12a}$$

$$\bar{C}_P = \frac{1}{T}\int_t^{t+T}\left[C_L \frac{\dot{y}(t)}{U_\infty} + C_M \frac{\dot{\alpha}(t)c}{U_\infty}\right] dt \tag{12b}$$

where Y_P is the difference between the highest and the lowest points reached by the foil, and $T = 1/f_e$ is the time period. Furthermore, the term C_P represents the power coefficient that can be decomposed as $C_P = C_P^p + C_P^\nu$. Here C_P^p denotes the power coefficient based on pressure only, whereas C_P^ν is the shear stress-based power coefficient.

For the foil pitching with an amplitude of $A_\alpha = 20^\circ$, we identify well-known operational regimes in the range of $0 < \mathrm{St}_A \leq 1.0$, including drag-production, thrust-production, and deflected wake regions (see Figure 3). Moreover, we observe that the neutral wake (i.e., the case for which the mean drag coefficient is zero) is formed at $\mathrm{St}_A = 0.18$. As the power generation efficiency of the foil is a concern, we show the variation in η in terms of a contour plot for the pitching amplitude A_α and Strouhal number St_A in Figure 4. In this parametric space, we obtain a remarkable power efficiency of 42%. For comprehensive details about the hydrodynamic performance and the wake topology of the flapping foil, the readers are referred to read the following article [40].

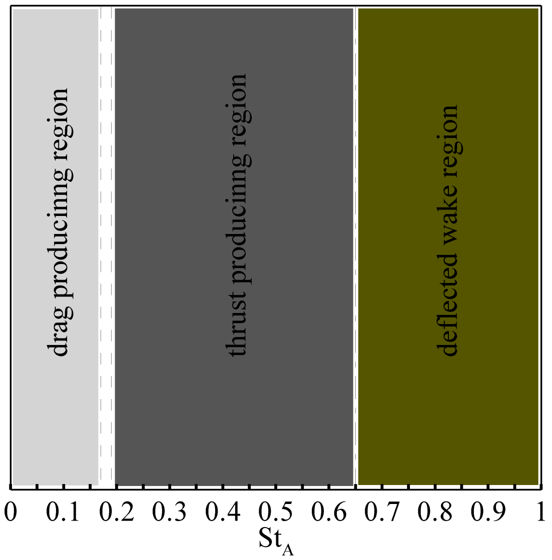


Figure 3. Operational regimes of flapping foil at a pitching amplitude of $A_\alpha = 20^\circ$.

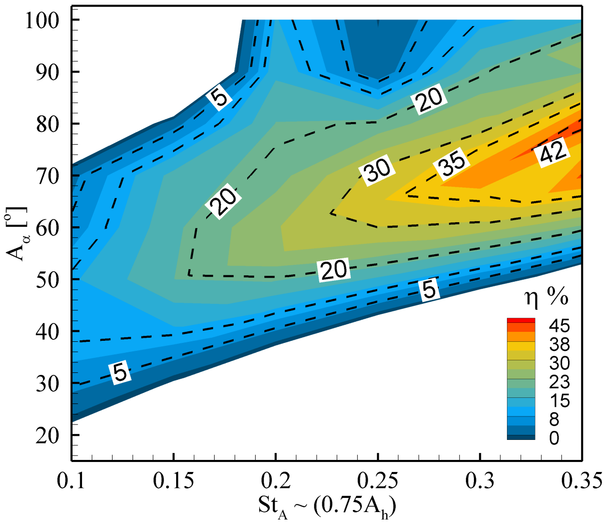


Figure 4. Contours of power generation efficiency in parametric space (St_A, A_α).

Next, we develop a ROM for the pressure profiles over the flapping foil, exhibiting power generation mode [41] through pressure-based POD and deep-learning neural network techniques. Our primary objective is to reconstruct the forces and moments exerted by the fluid on the body through the trained deep-learning model more efficiently. For this purpose, we choose a case of a foil flapping with A_α as 67° and $\mathrm{St}_A = 0.33$ (or $A_h = 0.44$), because these kinematic parameters correspond to a remarkable power generation efficiency of 40% at $\mathrm{Re} = 1100$ (see Figure 4). We simulate this case for a long enough time duration in order to mitigate the transient effects and obtain the periodic steady state. The time histories of the lift and the pitching moment coefficients are shown in Figure 5. We observe a phase of 90° between C_L and C_M. We present the snapshots of vorticity contours over a complete oscillation cycle in Figure 6. From these plots, we observe a pair of vortices, formed at the leading-edge during the first half-cycle, and similar coherent flow structures produced during the other half-cycle. These pairs then interacts with those formed at the trailing-edge, resulting in a dominant single vortex in the wake. As reported earlier, the leading edge vortex dominates the output of positive power generation [41,47], we notice a similar phenomenon here. Next, we discuss the strategies for computing pressure POD modes and training a deep-learning neural network model to develop ROMs for hydrodynamic forces and moments in the following passages.

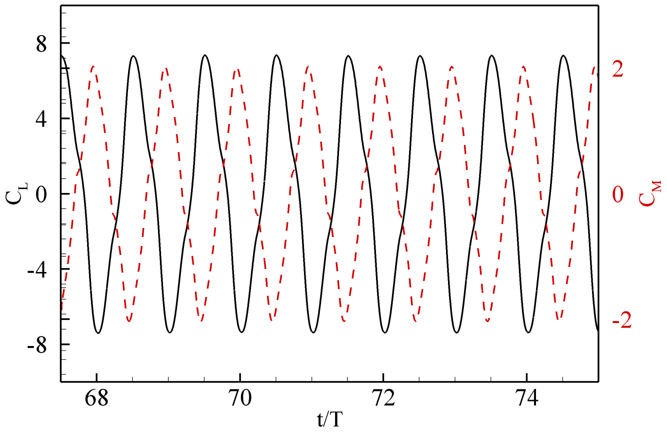


Figure 5. Time histories of lift (solid line) and pitching moment (dashed line) coefficients at $\mathrm{St}_A = 0.33$.

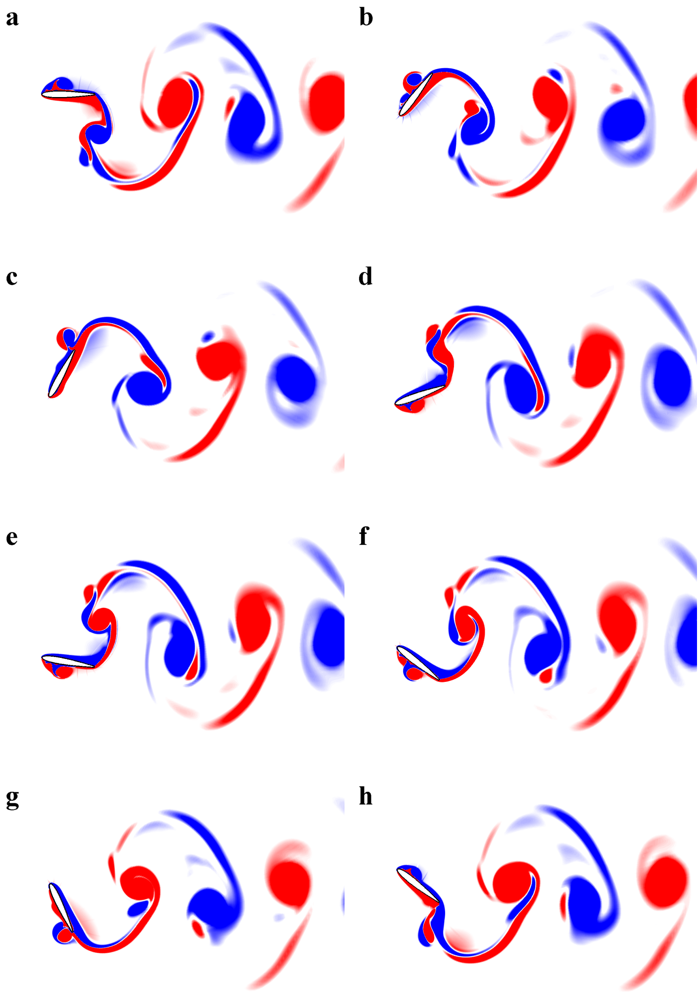


Figure 6. Spanwise vorticity contours over a complete oscillating cycle of flapping foil at $\mathrm{St}_A = 0.33$. Plots from one-half oscillating cycle are depicted in (**a**–**d**), which are corresponding to the time-instants from $t/T = 0$ to $3/8$ while the plots from the other half cycle are depicted in (**e**–**h**). Here, the symbol T represents the time period of an oscillating cycle.

5.1. Proper Orthogonal Decomposition

Here, we elaborate the methodology to compute dominant coherent structures of the flow field. One of the commonly known techniques is the POD method to extract dominant modes of the pressure field. Generally, the POD modes can be computed through the procedure of either SVD or the method of snapshots [68,69]. In this study, we utilize the method of snapshots because it is computationally inexpensive compared to SVD for high-fidelity simulations. The set of ensemble S time-discrete snapshots $(p_n = p(x, t_n)$, $t_n = t_s + (n-1)\Delta t, n = 1, 2, \cdots, S)$ is used as an input for POD of a flow property, such as pressure. Moreover, the time-dependent flow field p can be decomposed into its mean $\bar{p}(x)$ and fluctuating part $p'(x,t)$, and the fluctuating part $p'(x,t)$ is expanded in a Galerkin fashion in terms of temporal and spatial variables, as given below:

$$\begin{aligned} p(x,t) = \bar{p}(x) + p'(x,t) &= \bar{p}(x) + \sum_{j=1}^{\infty} a_j(t)\varphi_j(x,t) \\ &\approx \bar{p}(x) + \sum_{j=1}^{m} a_j(t)\varphi_j(x) \end{aligned} \tag{13}$$

where m represents the finite number of modes in the expansion, $\varphi(x)$ represents the POD modes of p to be determined through the eigenvalue problem, and $a_j(t)$ denotes the temporal coefficients. Note that, in this study, our primary aim is to propose and train a deep-learning model to estimate a_j. The instantaneous solutions at the time $t_1, t_2, \cdot\cdot, t_s \in (0, t)$ of P are stored in a matrix of size $N \times S$, where $S \ll N$, and N denotes the number of grid points. The problem is to seek a low dimensional basis $\{\varphi_1, \varphi_2 \cdot \cdot \varphi_m\}$ that satisfies the following relation:

$$\min \frac{1}{S} \sum_{i=1}^{s} \left\| p'(\cdot, t_i) - \sum_{j=1}^{m} \big(p'(\cdot, t_i), \varphi_j(\cdot)\big)_H \right\|_H^2 \tag{14}$$

subject to the condition of orthogonality, $(\varphi_i, \varphi_j)_H = \delta_{i,j}, 1 \leq i, j \leq m$, where $\delta_{i,j}$ is the Kronecker delta function and H is a real Hilbert space such that $p(\cdot, t) \in H \,|t \in (0, T)$. To solve Equation (14), we consider the following eigenvalue problem

$$Cv = \lambda v \tag{15}$$

where v_k for $k = 1, 2, \cdot \cdot \cdot, S$ are the eigenvectors, $\lambda_1 \geq \lambda_2 \cdot \cdot \cdot \geq \lambda_S > 0$ are eigenvalues, and $C \in R^{S \times S}$ is the correlation matrix that can be defined as

$$C_{ij} = \frac{1}{S} \big(p'(\cdot, t_j), p'(\cdot, t_i)\big)_H \tag{16}$$

It can be shown that the solution of Equation (14) is given by [68]

$$\varphi_k(\cdot) = \frac{1}{\sqrt{\lambda_k}} \sum_{j=1}^{S} (v_k)_j p'(\cdot, t_j), 1 \leq k \leq m \tag{17}$$

We compute the normalized eigenvalues ($\lambda_k / \sum_j \lambda_j$) to exhibit the quantification of variance/energy in the POD modes. In Figure 7, the pressure-based eigenvalues (λ^p) are shown along with the cumulative amount of energy. It can be inferred that the first 20 POD modes contain almost 99.98% of total energy. Consequently, the first 20 modes should be enough for a low-dimensional model of the pressure field, and hence, of the hydrodynamic forces and moment based on pressure only.

Figure 7 presents the contributions of the modes towards the energy of this nonlinear dynamical system. It also indicates that the first four modes carry 98% of the total energy. Unlike the circular cylinder [70] and static airfoils at different angles of attack [32], where pressure modes are either pairwise symmetric or antisymmetric, the modes in the case of flapping airfoil corresponding to the power generation regime do not exhibit pairwise symmetry or antisymmetry. It is observed that after a symmetric mode about the center line ($y = 0$), there is a pair of antisymmetric modes. In Figure 8, we show the mean and first four pressure POD modes. The mean mode (Mode 0) is symmetric as it shows a symmetric wake, causing zero mean lift, i.e., $\bar{C}_L = 0$). Here, we compute these POD modes using 200 snapshots per oscillating cycle in order to include small-scale coherent structures of pressure field in the computational process.

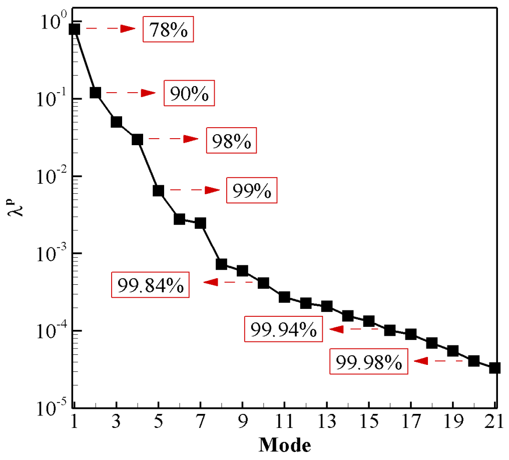


Figure 7. Normalized eigenvalues of the correlation matrix *C* and their corresponding contributions for the cumulative energy content.

5.2. Models of Hydrodynamic Forces and Moment

We develop the low-dimension models for fluidic forces (lift and drag) and moment (pitching moment) by projecting their respective full-dimensional models (given in Equation (9a)–(9c)) into space generated by pressure POD modes. It means that the modal forms of normal and axial, and pitching moment can be obtained by replacing the term p with its mode in Equation (13) as follows:

$$\begin{aligned} F_N^m &= -\oint_c \left(\bar{p}^s \sin(\theta) + \sum_{j=1}^{M} a_j(t)\varphi_j^s \sin(\theta) \right) ds \\ &= L_o^0 + \sum_{j=1}^{M} a_j(t) L_j^0 \end{aligned} \tag{18a}$$

$$\begin{aligned} F_A^m &= -\oint_c \left(\bar{p}^s \cos(\theta) + \sum_{j=1}^{M} a_j(t)\varphi_j^s \cos(\theta) \right) ds \\ &= D_o^0 + \sum_{j=1}^{M} a_j(t) D_j^0 \end{aligned} \tag{18b}$$

$$M^m = (D_o^1 - L_o^1) + \sum_{j=1}^{M} a_j(t)(D_j^1 - L_j^1) \tag{18c}$$

where

$$L_o^n = -\oint_c (x^b - x_p)^n \bar{p}^s \sin(\theta) ds, \tag{19a}$$

$$L_j^n = -\oint_c (x^b - x_p)^n \varphi_j^s \sin(\theta) ds, \tag{19b}$$

$$D_o^n = -\oint_c (y^b - y_p)^n \bar{p}^s \cos(\theta) ds, \tag{19c}$$

$$D_j^n = -\oint_c (y^b - y_p)^n \varphi_j^s \cos(\theta) ds, \tag{19d}$$

where the subscripts o and j represent the decomposition of $\bar{p}^s$ and φ_j^s into their sine and cosine components, respectively. The terms $\bar{p}^s$ and φ_j^s represent the data on the surface of the foil surface from the $\bar{p}$ and φ_j, respectively. Besides, x^b and y^b are the coordinates of the nodal points on the body surface, which are, in general, functions of arc-length and time in the case of flapping foil, i.e., $x^b = x^b(s,t)$ and $y^b = y^b(s,t)$. The exponent $'n'$ is an integer that is either 0 or 1. It is important to note that in the case of stationary bodies, terms L_j^0 and D_j^0 are called lift decomposition coefficient and drag decomposition coefficient, respectively, and make respective contributions to the lift and drag coefficients as the weight of the temporal coefficient [32,70].

Although the pressure temporal coefficient a_j is predicted from the trained ANN model. However, in the training phase, we need a_j *apriori*. To fulfill this requirement, we utilize orthogonality condition of the modes and derive the following relation in order to compute a_j corresponding to each centralized snapshot: $a_j = \left(p'(x,t_j), \varphi_j(x)\right)_H$. We use data from ten oscillating cycles of the periodic steady-state phase. Two hundred snapshots per cycle are recorded that makes a total of 2000 snapshots in ten cycles. Then, we utilize these temporal coefficients for training the ANN model, as explained in the next subsection.

5.3. Long-Short-Term Neural Network

In 1997, Sepp Hochreiter and Jurgen Schmidhuber proposed the use of LSTM networks as a deep-learning approach for time series data [71]. LSTM networks are a type of recurrent neural network (RNN) that are designed to address the problem of vanishing gradients [72]. A recurrent neural network can process sequential data, where the output of a previous step is used as an input for the next step. However, traditional RNNs suffer from the problem of the vanishing gradients, where a gradient used to update the weights becomes very small as it propagates through the layers, making it difficult for the network to learn. The basic purpose of using LSTM networks is to overcome this critical problem. They consist of an LSTM cell, which includes three gates: an input gate, an output gate, and a forget gate. These gates control the flow of information in the LSTM cell and help prevent from the vanishing gradient problem. The input gate controls the flow of new information into the cell, the output gate controls the flow of information out of the cell, and the forget gate controls what information should be forgotten from the cell. These gates and the states within the LSTM cell can be calculated using the following equations:

$$\text{Input gate: } i_t = \sigma(W_i \cdot [y_{t-1}, a_t] + b_i), \tag{20a}$$

$$\tilde{C}_t = \tanh(W_C \cdot [y_{t-1}, a_t] + b_C), \tag{20b}$$

$$\text{Forget gate: } f_t = \sigma(W_f \cdot [y_{t-1}, a_t] + b_f), \tag{20c}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{20d}$$

$$\text{Output gate: } o_t = \sigma(W_o \cdot [y_{t-1}, a_t] + b_o), \tag{20e}$$

$$y_t = o_t * \tanh(C_{t-1}). \tag{20f}$$

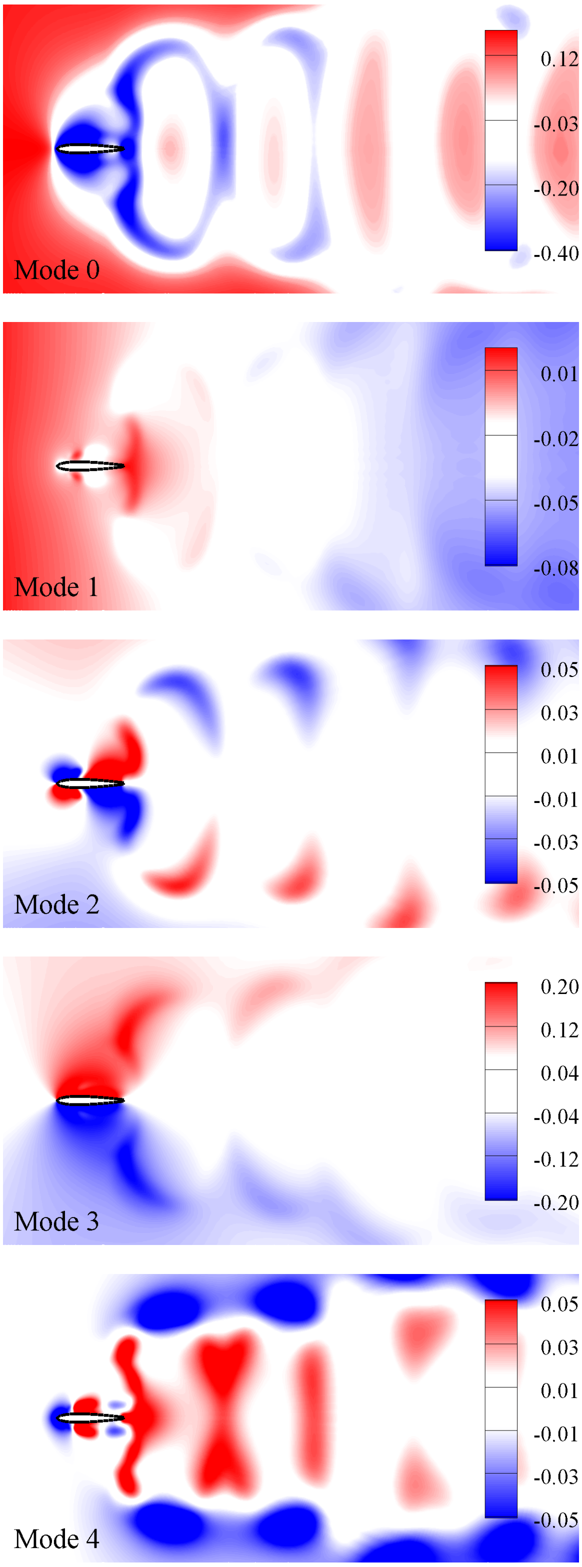


Figure 8. POD modes of pressure field at $\mathrm{St}_A = 0.33$.

The input of an LSTM cell is $[y_{t-1}, a_t]$, where y_{t-1} is the previous hidden state, and a_t is the current input. Each gate is calculated by applying a sigmoid function to the dot product of the input $[y_{t-1}, a_t]$ and the corresponding weight matrix plus bias. The sigmoid function outputs a value between 0 and 1.

The cell state C_t is calculated by multiplying the previous cell state with the forget gate value and adding the input gate value multiplied by the candidate cell state computed by applying a *tanh* function to the dot product of the input $[y_{t-1}, a_t]$ and the weight matrix W_c plus bias b_c. The *tanh* function produces a value between -1 and 1 as the output. Finally, the hidden state y_t is quantified by applying the output gate value to the *tanh* of the cell. TensorFlow, a python library for numerical computing [73], is utilized to train the neural network. The Adam optimizer; a method for updating the weights of the network, is employed during the training process. The Adam optimizer incorporates an exponential decay and learning rate adjustment in its governing mathematical equation, as described by [74].

The input layer of the LSTM consists of the temporal coefficients at time level 't'. The input data $\{a_1^{(t)}, a_2^{(t)}, ..., a_n^{(t)}\} \in \mathbb{R}^n$ is then routed via hidden layers followed by the *tanh* activation function. The output layer consists of the temporal coefficients at time level '$t+1$' $\{a_1^{(t+1)}, a_2^{(t+1)}, ..., a_n^{(t+1)}\} \in \mathbb{R}^n$. A schematic diagram to illustrate the training architecture is provided in Figure 9.

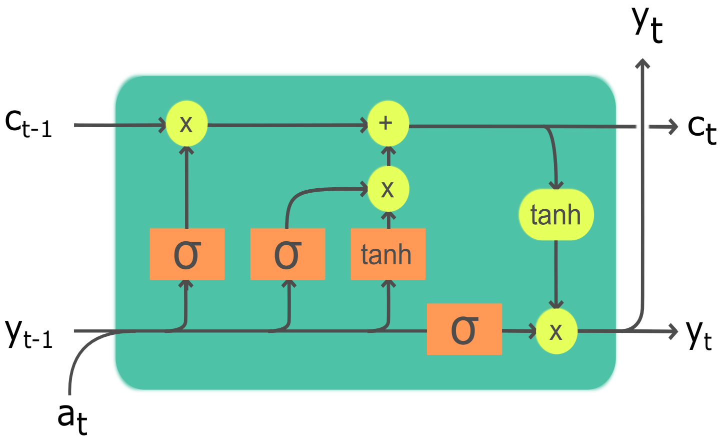


Figure 9. LSTM architecture with different gates. a represents the temporal coefficients where y represents the hidden state.

The data set is first divided into training and testing categories, with the training category making up 50% of the entire data set. The LSTM algorithm is then used to predict the next value in an initial sequence of data. The prediction process begins by providing the LSTM with the initial sequence of a length t and then, using the LSTM algorithm to predict the value at $t+1$. This prediction process is then repeated recursively, using the newly predicted value as an input for the next prediction. This process allows the LSTM to make predictions for future values in the sequence based on the patterns it learns from the training data.

In our present work, we employ 200 hidden units with four hidden layers with 2000 epoch. In addition, dropout is used to prevent overfitting. After training, the network is utilized as a function to predict future time steps using the initial condition very similar to a conventional ROM, where the set of initial-valued ordinary differential equations is integrated over time. Figure 10 presents time histories of the predicted temporal coefficients along with their true values. The dashed vertical line serves as a boundary to separate the training data and testing data. The data used for testing is not meant for training and is hidden from the trained network. The trained LSTM model efficiently approximates the pressure POD coefficients, as evident from the comparison in Figure 10. We use the predicted temporal coefficients in the low-dimensional models of hydrodynamic forces and moment (Equation (18a)–(18c) and reconstructed the pressure-based lift coefficient ($C_L^{\{m,p\}}$) and pitching moment coefficient ($C_M^{\{m,p\}}$) using ten POD modes. The comparison of time histories of $C_L^{\{m,p\}}$ and $C_M^{\{m,p\}}$ along with the data of direct numerical simulation (DNS) are presented in Figure 11. Moreover, we measure the accuracy of our model by computing the root-mean-square deviation (RMSD) between the predicted quantity and true DNS-based quantity. We notice that the RMSD values, in 10 oscillation cycles, containing both training and tested data (five cycles of each), are 13.76% and 8.26% in the lift and pitching moment coefficients, respectively. Thus, it exhibits 86.24% and 91.74% accuracy of the model for lift and pitching moment coefficients, respectively.

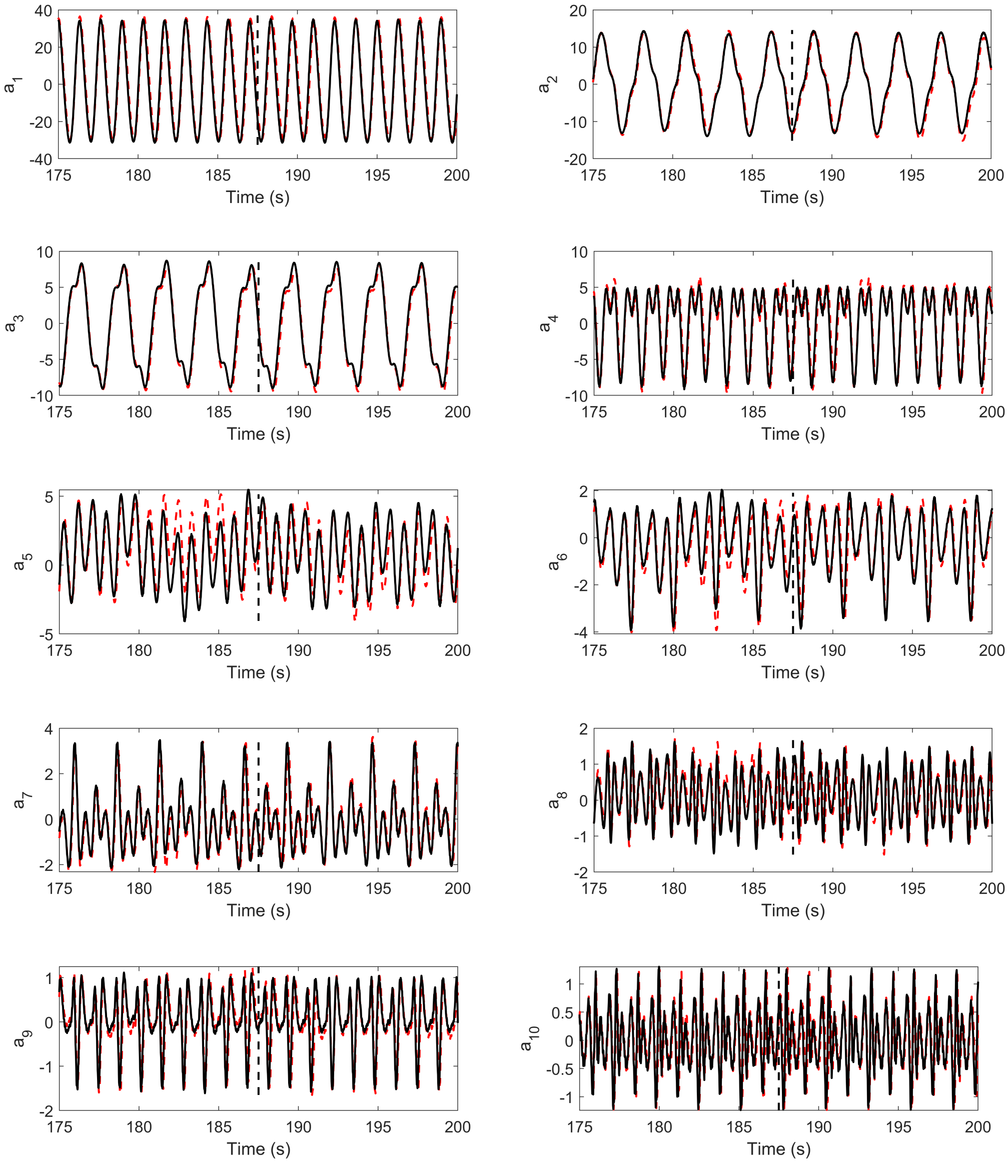
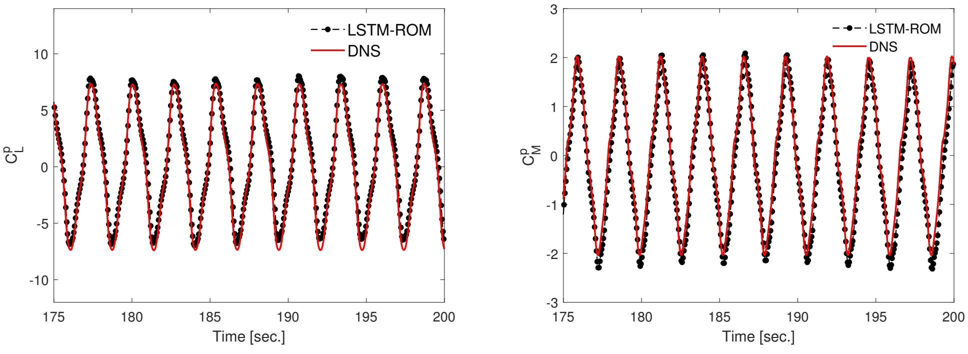


Figure 10. POD temporal coefficients: predicted (dashed line) and true (solid line). The vertical line splits the training and testing data.

It is important to note that the pressure temporal coefficients are not only required for the reconstruction of hydrodynamic forces but also they are very useful for the development of efficient ROMs for shear flows. Noack et al. [30] reported that POD-based ROM (or POD-ROM) of incompressible shear flows with the inclusion of pressure term improves the accuracy of POD-ROMs. They emphasized that the lack of pressure term resulted in an amplitude error that could not be compensated by simply increasing the number of modes. Besides, Tallet et al. [75] further explained the importance of pressure terms in the development of an efficient POD-ROM for shear flows. They utilized the projection of momentum equations to compute both the velocity and pressure temporal coefficients.

They examined the effectiveness of their methodology for POD-ROMs for a periodic flow past a circular cylinder and reconstructed the hydrodynamic coefficients as well as the Strouhal number, which provides a good agreement with those of the full-order model. Usually, projection of the pressure-Poisson equation is carried out, which results in a coupled velocity-pressure temporal coefficients system, and thus, it does not remain straightforward to be solved. This approach also stays limited to low-Reynolds number flows [21]. Moreover, [70] used the quadratic stochastic estimator to construct a relation between the temporal coefficients of pressure to the temporal coefficients of the velocity field through a mapping function. It helped us develop a ROM of the pressure field for flows over a cylinder at Re = 100 instead of using the conventional approach [21]. Thus, estimating the temporal coefficients (of either velocity or pressure, or both) using deep-learning neural networks provides an effective solution to handle serious deficiencies of conventional techniques for developing efficient, accurate, and robust ROMs. Furthermore, the proposed LSTM model is not limited to pressure coefficients only, velocity coefficients can also be appended in training. Although this addition increases the amount of the data set and hence, the training time along with the tuning of training parameters.



(**a**) (**b**)

Figure 11. Reconstruction of pressure-based (**a**) lift coefficient C_L^p and (**b**) moment coefficient C_M^p using ten modes: predicted and true.

6. Conclusions

In this study, a two-dimensional incompressible fluid flow around a NACA-0012 flapping foil is simulated and validated. The objective of this study is to develop a ROM for a power generating flapping foil using a deep neural networks model that works similarly to traditional ROMs. In the conventional ROMs of flow fields, the dynamical system along with the initial set of data, or precise initial conditions, are integrated over time for the prediction of future data, which may lead to erroneous results when prediction is required for a long-time duration. Here, we employ an LSTM network along with pressure POD to develop an efficient and a robust ROM for the pressure field over the surface of a flapping foil. The LSTM model is trained by temporal coefficients of a power generation system at St = 0.33 and $A_\alpha = 67°$, which correspond to almost 40% power generation efficiency of a flapping foil. It is also tested with the data that are not used in the training process. Furthermore, we gauge the robustness of the new LSTM-based model to predict the future temporal coefficients for a long time duration that is far beyond the training time interval. The results show a good efficiency and robustness of the proposed LSTM model, which can accurately predict the temporal coefficients and corresponding hydrodynamic forces and moment. It also demonstrates a good agreement with the true values obtained from the full-order simulation. We find 86.24% and 91.74% accuracy of the model in the reconstruction of lift and pitching moment coefficients, respectively. Therefore, the LSTM model can be a viable tool for the prediction of a flow field for a long-time duration. It is also more efficient and accurate compared to the conventional Galerkin projection-based POD-ROM techniques.

Author Contributions: Conceptualization, A.S., H.F. and I.A.; methodology, A.S., H.F. and I.A.; software, A.S., H.F., I.A., M.A.T. and M.S.U.K.; validation, H.F., I.A., Y.Y., M.A.T. and M.S.U.K.; formal analysis, A.S., H.F., I.A. and M.A.T.; investigation, A.S. and H.F.; resources, I.A.; writing—original draft preparation, A.S., H.F., I.A. and M.S.U.K.; writing—review and editing, H.F., I.A., M.A.T. and M.S.U.K.; visualization, A.S., H.F., I.A. and M.S.U.K.; supervision, I.A.; project administration, I.A.; funding acquisition, I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is supported by Digital Pakistan Lab under the National Center for Big Data and Cloud Computing funded by Higher Education Commission, Pakistan.

Data Availability Statement: Data can be provided by the authors upon reasonable requests.

Acknowledgments: The second author is thankful to the Computing Research and Development Center (CRDC) at the Institute of Computing, Bahauddin Zakariya University, Multan, Pakistan, for providing access to the SMP/multicore machine.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akhtar, I.; Borggaard, J.; Burns, J.A.; Imtiaz, H.; Zietsman, L. Using functional gains for effective sensor location in flow control: a reduced-order modelling approach. *J. Fluid Mech.* **2015**, *781*, 622–656. [CrossRef]
2. Buffoni, M.; Camarri, S.; Iollo, A.; Salvetti, M. Low-dimensional modelling of a conned three-dimensional wake ow. *J. Fluid Mech.* **2006**, *569*, 141150. [CrossRef]
3. Fang, F.; Pain, C.; Navon, I.; Gorman, G.; Piggott, M.; Allison, P.; Farrell, P.; Goddard, A. A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. *Ocean. Model.* **2009**, *28*, 127–136. [CrossRef]
4. Fortuna, L.; Nunnari, G.; Gallo, A. *Model Order Reduction Techniques with Applications in Electrical Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
5. Noack, B.R.; Morzynski, M.; Tadmor, G. *Reduced-Order Modelling for Flow Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 528.
6. Bakewell, H.P.; Lumley, J.L. Viscous sublayer and adjacent wall region in turbulent pipe flow. *Phys. Fluids* **1967**, *10*, 1880–1889. [CrossRef]
7. Deane, A.; Kevrekidis, I.; Karniadakis, G.E.; Orszag, S. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Phys. Fluids Fluid Dyn.* **1991**, *3*, 2337–2354. [CrossRef]
8. Shirovich, L. Turbulence and the Dynamics of Coherent Structures; Part I: Coherent Structures. *Q. Appl. Math.* **1987**, *45*, 561–571. [CrossRef]
9. Noack, B.R.; Schlegel, M.; Ahlborn, B.; Mutschke, G.; Morzyński, M.; Comte, P. A finite-time thermodynamics of unsteady fluid flows. *J. Non-Equilib. Thermodyn.* **2008**, *33*, 103–148. [CrossRef]
10. Graham, W.R.; Peraire, J.; Tang, K.Y. Optimal control of vortex shedding using low order models. Part I: Open-loop model development. *Int. J. Numer. Methods Eng.* **1999**, *44*, 945–972. [CrossRef]
11. Graham, W.R.; Peraire, J.; Tang, K.Y. Optimal control of vortex shedding using low order models. Part II: Model-based control. *Int. J. Numer. Methods Eng.* **1999**, *44*, 973–990. [CrossRef]
12. Akhtar, I.; Nayfeh, A.H. Model based control of laminar wake using fluidic actuation. *J. Comput. Nonlinear Dyn.* **2010**, *5*, 041015. [CrossRef]
13. Kunisch, K.; Volkwein, S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optim. Theory Appl.* **1999**, *102*, 345–371. [CrossRef]
14. Yue, Y. The Use of Model Order Reduction in Design Optimization Algorithms. Ph.D. Thesis, Katholieke Univerisiteit, Leuven, Belgium, 2012.
15. Imtiaz, H.; Akhtar, I. Nonlinear closure modeling in reduced order models for turbulent flows: A dynamical system approach. *Nonlinear Dyn.* **2020**, *99*, 479–494. [CrossRef]
16. Wang, Z.; Akhtar, I.; Borggaard, J.; Iliescu, T. Two-Level Discretizations of Nonlinear Closure Models for Proper Orthogonal Decomposition. *J. Comput. Phys.* **2011**, *230*, 126–146. [CrossRef]
17. Wang, Z. Reduced-order modeling of complex engineering and geophysical flows: analysis and computations. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2012.
18. Östh, J.; Noack, B.R.; Krajnović, S.; Barros, D.; Borée, J. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *J. Fluid Mech.* **2014**, *747*, 518–544. [CrossRef]
19. Xiao, D.; Fang, F.; Buchan, A.G.; Pain, C.C.; Navon, I.M.; Du, J.; Hu, G. Non-linear model reduction for the Navier–Stokes equations using residual DEIM method. *J. Comput. Phys.* **2014**, *263*, 1–18. [CrossRef]
20. Sabetghadam, F.; Jafarpour, A. α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *Appl. Math. Comput.* **2012**, *218*, 6012–6026. [CrossRef]
21. Akhtar, I.; Nayfeh, A.H.; Ribbens, C.J. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theor. Comput. Fluid Dyn.* **2009**, *23*, 213–237. [CrossRef]

22. Chaturantabut, S.; Sorensen, D.C. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764. [CrossRef]
23. Schlegel, M.; Noack, B.R. On long-term boundedness of Galerkin models. *J. Fluid Mech.* **2015**, *765*, 325–352. [CrossRef]
24. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [CrossRef]
25. Kochkov, D.; Smith, J.A.; Alieva, A.; Wang, Q.; Brenner, M.P.; Hoyer, S. Machine learning–accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2101784118. [CrossRef] [PubMed]
26. Kutz, J.N. Deep learning in fluid dynamics. *J. Fluid Mech.* **2017**, *814*, 1–4. [CrossRef]
27. Guo, M.; Hesthaven, J.S. Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* **2019**, *345*, 75–99. [CrossRef]
28. Han, J.; Jentzen, A.; Weinan, E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 8505–8510. [CrossRef] [PubMed]
29. Hesthaven, J.S.; Ubbiali, S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.* **2018**, *363*, 55–78. [CrossRef]
30. Noack, B.R.; Papas, P.; Monkewitz, P.A. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.* **2005**, *523*, 339–365. [CrossRef]
31. Ahmed, H.F.; Farooq, H.; Akhtar, I.; Bangash, Z. Machine learning–based reduced-order modeling of hydrodynamic forces using pressure mode decomposition. *Proc. Inst. Mech. Eng. Part J. Aerosp. Eng.* **2021**, 235, 2517–2528. [CrossRef]
32. Farooq, H.; Saeed, A.; Akhtar, I.; Bangash, Z. Neural Network-Based Model Reduction of Hydrodynamics Forces on an Airfoil. *Fluids* **2021**, *6*, 332. [CrossRef]
33. Fish, F.; Lauder, G. Passive and active flow control by swimming fishes and mammals. *Annu. Rev. Fluid Mech.* **2006**, *38*, 193–224. [CrossRef]
34. Rozhdestvensky, K.V.; Ryzhov, V.A. Aerohydrodynamics of flapping-wing propulsors. *Prog. Aerosp. Sci.* **2003**, *39*, 585–633. [CrossRef]
35. Triantafyllou, M.S.; Techet, A.H.; Hover, F.S. Review of experimental work in biomimetic foils. *IEEE J. Ocean. Eng.* **2004**, *29*, 585–594. [CrossRef]
36. Wang, Z.J. Dissecting insect flight. *Annu. Rev. Fluid Mech.* **2005**, *37*, 183–210. [CrossRef]
37. Guerrero, J. Numerical simulation of the unsteady aerodynamics of flapping flight. Ph.D. Thesis, Department of Civil, Environmental, Architectural Engineering Universita degli Studi di Genova, Genoa, Italy, 2009.
38. Xiao, Q.; Zhu, Q. A review on flow energy harvesters based on flapping foils. *J. Fluids Struct.* **2014**, *46*, 174–191. [CrossRef]
39. Young, J.; Lai, J.C.; Platzer, M.F. A review of progress and challenges in flapping foil power generation. *Prog. Aerosp. Sci.* **2014**, *67*, 2–28. [CrossRef]
40. Farooq, H.; Ghommem, M.; Khalid, M.S.U.; Akhtar, I. Numerical investigation of hydrodynamic performance of flapping foils for energy harvesting. *Ocean. Eng.* **2022**, *260*, 112005. [CrossRef]
41. Kinsey, T.; Dumas, G. Parametric study of an oscillating airfoil in a power-extraction regime. *AIAA J.* **2008**, *46*, 1318–1330. [CrossRef]
42. Platzer, M.; Ashraf, M.; Young, J.; Lai, J. Extracting power in jet streams: Pushing the performance of flapping wing technology. In Proceedings of the 27th Congress of the International Council of the Aeronautical Sciences, International Council of the Aeronautical Sciences Paper, Nice, France, 19–24 September 2010; Volume 2.
43. Ashraf, M.; Young, J.; Lai, J.; Platzer, M. Numerical analysis of an oscillating-wing wind and hydropower generator. *AIAA J.* **2011**, *49*, 1374–1386. [CrossRef]
44. Zhu, Q. Energy harvesting by a purely passive flapping foil from shear flows. *J. Fluids Struct.* **2012**, *34*, 157–169. [CrossRef]
45. Shimizu, E.; Isogai, K.; Obayashi, S. Multiobjective design study of a flapping wing power generator. *J. Fluids Eng.* **2008**, *130*. [CrossRef]
46. Zhu, Q.; Haase, M.; Wu, C.H. Modeling the capacity of a novel flow-energy harvester. *Appl. Math. Model.* **2009**, *33*, 2207–2217. [CrossRef]
47. Zhu, Q.; Peng, Z. Mode coupling and flow energy harvesting by a flapping foil. *Phys. Fluids* **2009**, *21*, 033601. [CrossRef]
48. Abiru, H.; Yoshitake, A. Study on a flapping wing hydroelectric power generation system. *J. Environ. Eng.* **2011**, *6*, 178–186. [CrossRef]
49. Wu, J.; Chen, Y.; Zhao, N. Role of induced vortex interaction in a semi-active flapping foil based energy harvester. *Phys. Fluids* **2015**, *27*, 093601. [CrossRef]
50. Deng, J.; Teng, L.; Pan, D.; Shao, X. Inertial effects of the semi-passive flapping foil on its energy extraction efficiency. *Phys. Fluids* **2015**, *27*, 053103. [CrossRef]
51. Farooq, H.; Khalid, M.S.U.; Akhtar, I.; Hemmati, A. Nonlinear response of passively flapping foils. *Ocean. Eng.* **2022**, *261*, 112071. [CrossRef]
52. Young, J.; Ashraf, M.A.; Lai, J.C.; Platzer, M.F. Numerical simulation of fully passive flapping foil power generation. *AIAA J.* **2013**, *51*, 2727–2739. [CrossRef]
53. Veilleux, J.C.; Dumas, G. Numerical optimization of a fully-passive flapping-airfoil turbine. *J. Fluids Struct.* **2017**, *70*, 102–130. [CrossRef]

54. Wang, Z.; Du, L.; Zhao, J.; Sun, X. Structural response and energy extraction of a fully passive flapping foil. *J. Fluids Struct.* **2017**, *72*, 96–113. [CrossRef]
55. Ashraf, M.A. Numerical simulation of the flow over flapping airfoils in propulsion and power extraction regimes. Ph.D. Thesis, University of New South Wales, Australia, 2010.
56. Daqaq, M.F.; Bibo, A.; Akhtar, I.; Alhadidi, A.H.; Panyam, M.; Caldwell, B.; Noel, J. Micropower generation using cross-flow instabilities: A review of the literature and its implications. *J. Vib. Acoust.* **2019**, *141*, 030801. [CrossRef]
57. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
58. San, O.; Maulik, R.; Ahmed, M. An artificial neural network framework for reduced order modeling of transient flows. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *77*, 271–287. [CrossRef]
59. Saeed, A.; Farooq, H.; Akhtar, I.; Bangash, Z. Deep learning-based reduced-order model for turbulent flows. In Proceedings of the 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST), IEEE, Islamabad, Pakistan, 16–20 August 2022; pp. 821–829.
60. Wang, Z.; Xiao, D.; Fang, F.; Govindan, R.; Pain, C.C.; Guo, Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int. J. Numer. Methods Fluids* **2018**, *86*, 255–268. [CrossRef]
61. Gupta, R.; Jaiman, R. Three-dimensional deep learning-based reduced order model for unsteady flow dynamics with variable Reynolds number. *Phys. Fluids* **2022**, *34*, 033612. [CrossRef]
62. Pawar, S.; Ahmed, S.E.; San, O.; Rasheed, A. Data-driven recovery of hidden physics in reduced order modeling of fluid flows. *Phys. Fluids* **2020**, *32*, 036602. [CrossRef]
63. Zhang, X.; Ji, T.; Xie, F.; Zheng, H.; Zheng, Y. Unsteady flow prediction from sparse measurements by compressed sensing reduced order modeling. *Comput. Methods Appl. Mech. Eng.* **2022**, *393*, 114800. [CrossRef]
64. de Boer, A.; van der Schoot, M.; Bijl, H. Mesh deformation based on radial basis function interpolation. *Comput. Struct.* **2007**, *85*, 784–795. [CrossRef]
65. Bos, F.M.; van Oudheusden, B.W.; Bijl, H. Radial basis function based mesh deformation applied to simulation of flow around flapping wings. *Comput. Fluids* **2013**, *79*, 167–177. [CrossRef]
66. Akhtar, I. Parallel simulations, reduced-order modeling, and feedback control of vortex shedding using fluidic actuators. Ph.D. Thesis, Virginia Tech, Blacksburg, VA, USA, 2008.
67. Leonard, B.P. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Mech. Eng.* **1979**, *19*, 59–98. [CrossRef]
68. Sirovich, L. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. *Q. Appl. Math.* **1987**, *45*, 583–590. [CrossRef]
69. Sirovich, L.; Kirby, M. Low-dimensional procedure for the characterization of human faces. *JOSA A* **1987**, *4*, 519–524. [CrossRef]
70. Imtiaz, H.; Akhtar, I. On lift and drag decomposition coefficients in a model reduction framework using pressure-mode decomposition (PMD) analysis. *J. Fluids Struct.* **2017**, *75*, 174 – 192. [CrossRef]
71. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
72. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
73. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
74. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
75. Tallet, A.; Allery, C.; Leblond, C.; Liberge, E. A minimum residual projection to build coupled velocity–pressure POD–ROM for incompressible Navier–Stokes equations. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *22*, 909–932. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.