



Parameter optimization of the bio-inspired robot propulsion through the deep learning based reduced order fluid-structure interaction model

Zixiang Ying ^a, Linxiang Wang ^{a,*}, Roderick Melnik ^b

^a State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, 310027, Hangzhou, PR China

^b MS2Discovery Interdisciplinary Research Institute, Wilfrid Laurier University, Waterloo, ON, N2L 3L5, Canada

ARTICLE INFO

Keywords:

Bio-inspired locomotion
Fluid-structure interaction
Reduced order modeling
Proper orthogonal decomposition
Neural networks

ABSTRACT

In this paper, an effective model (POD-NIROM) is proposed, which makes full use of Long short-term memory Neural Network (LSTM NN) and proper orthogonal decomposition (POD) to predict the fluid dynamics around the moving boundary, as well as the soft robot locomotion. This is the first time that the proposed model has been used to optimize the propulsion parameters of the robotic fish, in which the body stiffness of the robot has been taken into consideration. To discuss the reliability of the presented model, the prediction and simulation of robot swimming performance are compared with experimental measurements, the results show a consistent trend. Finally, the trained model is used to optimize the robot's propulsion parameters. The results show that the robot has the best propulsion speed and propulsion force when the dimensionless wavenumber \tilde{k} is about 0.7. Compared with the high-fidelity model, the average relative standard deviation of the present model is 2.60%, but the calculation cost is reduced by 99.4%. Reasonable prediction and efficient calculation make the proposed POD-NIROM has great potential in the process of bio-inspired robot swimming prediction or bio-inspired robot propulsion parameter optimization.

1. Introduction

The locomotion of fish has the advantage of being fast, flexible and noiseless, which attracts great interest in submarine robotics design (Triantafyllou and Triantafyllou, 1995). Underwater bionic technology has received attention and has become one of the important research directions of underwater vehicles, such as snake-like robots (Perez et al., 2014), jellyfish-inspired robots (Christianson et al., 2019), robotic fish (Bergmann and Iollo, 2011; Katschmann et al., 2014), surface-swimming robots (Long et al., 2011a, 2011b), dolphin-like robot (Shen et al., 2013). However, the propulsion efficiency of the current bionic robotic fish is significantly lower than that of natural fish (Nguyen et al., 2012), which has aroused widespread attention to the optimization of underwater robot propulsion efficiency.

The computational fluid dynamics (CFD) method is widely used to predict the propulsion motion of robotic fish or the flow field around the robot (Bergmann and Iollo, 2011; Phamduy et al., 2016). Generally, a high-fidelity model using the CFD method can obtain a convincing solution when predicting the relationship between the fluctuation of the underwater vehicle and the external propulsion movement. Quick

response is necessary for models applied to optimization or path planning (Perez et al., 2014; Marchese et al., 2014). However, the nonlinearity of these equations, accompanied by high computational cost, makes it difficult for the model to achieve fast simulation.

A commonly used method to reduce computational cost in the simulation of soft underwater robots is to ignore the influence of fluid resistance on the lateral fluctuation amplitude of the robot. In other words, simplify the robot's swimming to rigid body motion (Bergmann and Iollo, 2011; Borazjani and Sotiropoulos, 2009). However, biological studies have shown that the internal parameters of underwater robots, especially body stiffness, strongly affect their swimming performance (Tytell et al., 2010). In contrast, the solution of a high-fidelity model that considers the stiffness of the robot's body is more valuable.

The reduced order model (ROM) is a commonly used method to obtain a reasonable approximation by constructing a low-dimensional subspace. The ROM can efficiently simulate physical and dynamic systems while maintaining reasonable accuracy, more specifically, retaining the main characteristics of the solution state vector (Schilders et al., 2008). These characteristics are sufficient to describe soft robot swimming, which is a two-way fluid-solid coupling problem.

* Corresponding author.

E-mail address: wanglx236@zju.edu.cn (L. Wang).

The ROM has been applied to various fields, for example, wake flows (Phamduy et al., 2016; Stankiewicz et al., 2016), shallow water (Răzvan et al., 2014; Lozovskiy et al., 2016), and ocean modeling (Fang et al., 2013; Z et al., 2016). Proper orthogonal decomposition (POD) is one of the most commonly used methods in building the ROM. It is a modal decomposition method designed to identify the most important flow features used to reconstruct snapshots (Tu et al., 2013). POD is used firstly in the field of fluid mechanics, to find the most contributing POD modes (a POD mode includes the POD space mode and the corresponding POD temporal coefficient) in a turbulent flow (Lumley, 1967). Since then, the POD method has been widely used in various applications of fluid mechanics, such as swirling flow Applications (Dragomirescu et al., 2013), control of the Burgers equation (Perrin et al., 2006), rarefied gas flows (Zhao et al., 2020), and turbulent flow in a square duct (Reichert et al., 1994). Simultaneously, it is a mature technique using the POD method to extract the characteristics of the Karman vortex street in the case of the cylinder wake (Zhao et al., 2019; Erwan and Hamdouni, 2010; Erwan et al., 2007).

The POD-Galerkin method is a POD-based intrusive ROM (POD-IROM) using Galerkin Projection to project governing equations to obtain ordinary differential governing equations (Eivazi et al., 2020). And new cases with varied parameter combinations (varied initial conditions) can be predicted. When using POD-IROM, many physical characteristics are retained, but it also brings challenges to model stability and nonlinear efficiency (Schlegel et al., 2015). POD-IROM is bound to the control equations and a new model must be established if the control strategy changes (Xiao et al., 2013). Fortunately, the POD-based non-intrusive ROM (POD-NIROM) has been proposed and used for flow field control (Kherad et al., 2020). The predictability of NIROM is achieved by a neural network framework. When optimizing the parameters for controlling the robot undulation, it is necessary to predict the underwater motion of the robot under various initial conditions. The prediction function of the POD-Galerkin method is realized by the ordinary differential governing equations. Once the initial conditions are changed, the coefficient matrixes of the equations need to be reassembled. In contrast, POD-NIROM has a lower computational cost, considering that, in POD-NIROM, different initial conditions correspond to different input signals and the neural network framework itself is unchanged.

Recently, efficient algorithms based on deep learning (Videler and Wardle, 1991) to predict fluid evolution have emerged. DL has been used to learn airflow characteristics on fixed aerodynamic structures for geometric optimization (Storti et al., 2019). In addition, deep learning is also used for Reynolds average turbulence modeling (Ling et al., 2016) and active flow control strategies (Rabault et al., 2019). Among them, long and short-term memory (LSTM), as a type of DL, has shown attractive potential in the modeling of fluid mechanics (Srinivasan et al., 2019) and has been successfully deployed to construct the NIROM (Mohan and Gaitonde, 2018). In brief, as the critical technology of the NIROM, DL can enhance the advantages of the ROM in shape optimization and flow control. In the POD-NIROM proposed in this paper, the LSTM framework plays the role of predicting the evolution of the POD temporal coefficients, taking the initial displacement of the robot (without considering hydrodynamics yet) as input.

Although POD-NIROM has great potential, its application in the prediction of bio-inspired robot swimming performance is still limited. Currently, this algorithm is very popular in aerodynamics researches, such as unsteady aerodynamic and aeroelastic modeling (Lee et al., 2019), inviscid transonic flow past an airfoil (Renganathan et al., 2020), aero-elastic interaction (Halder et al., 2020). It is worth mentioning that the theory of fish swimming and the theory of airfoil vibration have something in common. For example, the proposal of the kinematics for carangiform and anguilliform swimmers was inspired by the slender body theory in the field of aerodynamics (Wu, 1961). As has been proven effective in aerodynamics, POD NIROM is also seen as a promising avenue to predict the performance of swimmers.

In this work, a fast and effective algorithm is introduced to optimize the driving parameters of the underwater robot. The main idea of this algorithm is to use the POD method to reconstruct the dimensionality reduction model which is used to solve the bidirectional fluid-structure coupling problem of soft underwater vehicle motion and use LSTM to predict the POD modes used to reconstruct the model. In addition, the stiffness of the robot is taken into consideration, when predicting the swimming of the robot, which is a parameter that affects the swimming performance of the robot. Here, the robot is regarded as a uniform linear elastic body, and its elastic modulus reflects the difference between the actual swimming amplitude of the robot and the input signal. Finally, the trained model is used to optimize the robot's propulsion parameters, where the forward swimming speed and the propulsion force are the optimization indicators.

The rest of this article is organized as follows. Numerical methods used to predict robot swimming performance are introduced in Section 2. In Section 3, the details of model construction, as well as several important parameters and indicators are presented. In Section 4, the experimental measurement, HF model simulation, and present model prediction are compared to verify the reliability of the present model, and the robot swimming performance, as well as the swimming efficiency, are discussed.

2. Methodology

2.1. Dataset and POD

In this work, a 2D high-fidelity model (HF model) is considered to simulate the incompressible fluid around the moving deformable body. As discussed in existing research, the fish-like swimmer will turn (or the robot does not swim along the negative x-axis, in current cases) when starting up (in the form of the S-start) from rest (Gray, 1953). Additionally, fish have a good orientation when swimming upstream (Webb, 1984). To keep the robot swimming in a straight line, in the current HF model, the robot starts up against the current with a constant fluid velocity $U = 0.1\text{m/s}$. The start up process lasted for 0.5s, which means $U = 0$ when $t \geq 0.5\text{s}$. In the discussion of the following sections, the swimming cycles are counted after the start up process (e.g. $[0T - 1T]$ corresponds to $[0.5\text{s}, 1.5\text{s}]$, $U = 0$ for all swimming cycles). The boundary conditions on the outer boundary are uniform inflow at the inlet (the inlet normal velocity U), the sliding wall on the side boundary (no-slip boundary condition applied), and the convection boundary conditions at the outlet. The computational domain (Fig. 1 (B)) is a $10L \times 2L$ (L is the body length of the robot) rectangle with 15267 nodes and 27707 triangular elements. Fig. 1 shows a schematic description of the flow configuration. In Fig. 1, $\Omega = \Omega_f \cup \Omega_l$ is the entire computational domain. Ω_f is the fluid domain. Ω_l is the domain of the self-propelling object (fish-like swimmer) that provides the undulation motion d^0 regarded as the initial grid displacement upon the moving boundary of the fluid. For simplicity, Ω will be taken to be a rectangle, with an outer boundary $\Gamma_0 = \Gamma_i \cup \Gamma_o \cup \Gamma_{w1} \cup \Gamma_{w2}$. $\mathbf{n}_i, \mathbf{n}_o, \mathbf{n}_{w1}, \mathbf{n}_{w2}$ are the outward normal of Ω , \mathbf{n}_l is the outward normal of Ω_l .

In the high-fidelity model, the fluid is governed by the incompressible Navier-Stokes equations described by Arbitrary Lagrangian-Eulerian (ALE) formulation:

$$\rho_f \ddot{\mathbf{u}} + \rho_f (\mathbf{u} \cdot \nabla) \mathbf{u} - \dot{\mathbf{q}} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad \text{in } \Omega_f \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega_f \quad (2)$$

where \mathbf{u} is the flow velocity, and \mathbf{q} is the grid displacement. The fractional step method (Chorin, 1967; Zhao et al., 2007; Bao et al., 2011) is used for the time discretization of the momentum equation (Eq. (1)). The Streamline upwind Petrov-Galerkin finite element method (Brooks and Hughes, 1982; Choi and Yoo, 1997) is used for spatial discretization of the momentum equation. The mesh displacement of the fluid domain (\mathbf{q}

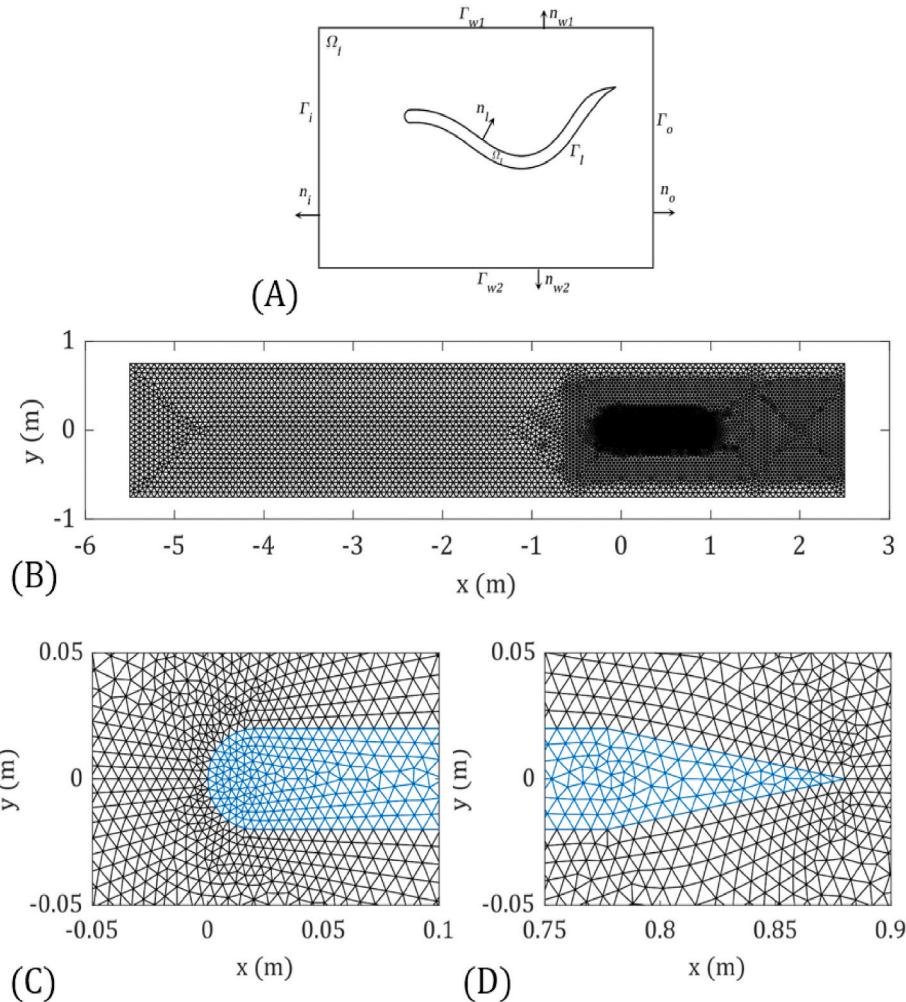


Fig. 1. (A) Sketch of the computational domain. (B) Mesh division of the computational domain. (C)–(D) Zoom-in view of the mesh near the head and tail, where the fluid and the robot are represented in black and blue, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

in Ω_f) is described by a modified version of the Laplace mesh motion algorithm (Masud and Hughes, 1997; Masud et al., 2007; Wang et al., 2014). The mesh displacement of the robot motion (q in Ω_I) is consistent with the robot displacement (u_{rob} in Ω_I). And the displacement of the robot (u_{rob} in Ω_I) is obtained by calculating the equilibrium equation of the linear elastic body:

$$\rho_I \ddot{u}_{rob} = \nabla \cdot \sigma_I + F_{ex} \text{ in } \Omega_I \quad (3)$$

$$\sigma_I \cdot n_I = (-\nabla p + \mu \nabla \cdot (\nabla u + \nabla u^T)) \cdot n_I \text{ on } \Gamma_I \quad (4)$$

where ρ_I is the density of the self-propelling object, σ_I is the Cauchy stress tensor, $F_{ex}(t) = D \nabla d^0(t)$ represents the time-varying external load (which is the energy source that drives the undulations of the robot), D is the elasticity matrix containing the material properties for Ω_I , $d^0(t)$ is the desired shape motion of the robot specified in Eq. (4). When solving Eq. (3), the Newton-Raphson method is introduced for iteration (Deuflhard, 2006) and the Generalized Hooke's Law is used for the construction of the linear elastic material model (Necas and Hlavacek, 1981). For simplicity, the elasticity of the soft robot body or joints is represented in the current model as the average body stiffness E and the Poisson's ratio ν .

The solution state vector $s(x, t)$ for the HF model (describing the fluid-solid interaction problem) is obtained by using the staggered algorithm (Wang et al., 2014; Mohamed et al., 2021; Placzek et al., 2009;

Persillon and Braza, 1998; Farhat et al., 1997). The steps of this algorithm can be described as: (i) update the boundary condition of the robot (Eq. (4)), based on the solution state vector $s(t_n)$ at t_n ; (ii) update the external load $F_{ex}(t_n)$ at t_n (Eq. (5)); (iii) calculate the displacement of the robot $u_s(t_{n+1})$ by solving Eq. (3); (iv) update the mesh configuration $q(t_{n+1})$ based on the mesh motion algorithm; (v) solve the Navier-Stokes equations (Eqs. (1) and (2)) on the updated mesh configuration to get the solution state vector $s(t_{n+1})$ at t_{n+1} ; (vi) return to (i). The current HF model is executed with the software FreeFem++ (Hecht, 2012). MATLAB is used for LSTM network model building (Kim and Kim, 2017) and data visualization.

The instantaneous axis of the robot is a traveling wave with increasing amplitude from fish head to fishtail (Gray, 2020). As described in the Lighthill propulsion model (Videler and Hess, 1984), the desired shape motion of the robot $d^0(t) = (0, d_y^0)$ is given as follows:

$$d_y^0(\xi, t) = A(\xi) \sin(k\xi + \sigma t) \quad (5)$$

where ξ is the axial length measured along the instantaneous axis (also known as a local coordinate, with $\xi = 0$ at the head and $\xi = L$ at the tail), $A(\xi)$ is the envelope function defining the maximum lateral displacement, k is the wavenumber, and σ is the angular frequency. Some coefficients are defined as follows:

$$A(\xi) = A_0 + A_1 \xi + A_2 \xi^2 \quad (6)$$

$$k = 2\pi\tilde{k}$$
(7)

$$\sigma = 2\pi f$$
(8)

$$\tilde{k} = \frac{L}{\lambda}$$
(9)

where the coefficients $A_0 = 0.1512$, $A_1 = 0$ and $A_2 = 0$ are defined to match the cases where the desired oscillating amplitude of each joint is 35° (Perez et al., 2014), the result is shown in Fig. 2(A) (the body wave fitting technique is used to estimate swimming kinematic parameters matched to the oscillation of the link-based body (Liu and Hu, 2010; Yu et al., 2004)), \tilde{k} is the dimensionless wavenumber, $f = 0.8$ is the undulating frequency of the robot, $L = 0.88\text{m}$ is the body length of the robot, λ is the wavelength of the body undulations. Furthermore, an appropriate robot stiffness E (details in Section 2.3) is chosen in the HF model to make the simulation consistent with the experiment. Intuitively, the envelope of the desired shape motion of the robot $A(\xi)$ (also the input of the POD-NIROM) and the envelope of the robot's lateral undulation are shown in Fig. 2(B).

Consider a space $\Omega \subset \mathbb{R}^d$ with $D = 1, 2, 3$, $x \in \Omega$. D is the dimension tied to space. And the time is denoted by $t \in T$, where $T \subset \mathbb{R}$. In a Hilbert space H , the solution state vector $s(x, t) \in H(\Omega, T)$, which includes the flow state variables and grid displacement, is solved by the HF model simulation. The solution state vector $s(x, t)$ and the standard inner product of $s(x, t)$ are given by:

$$s(x, t) = \begin{bmatrix} q_i(x, t) \\ u_i(x, t) \\ p(x, t) \end{bmatrix} \quad i = 1, 2 \quad (10)$$

$$(s(x_1, t_1), s(x_2, t_2)) = \int_{\Omega} s(x_1, t_1) \cdot s(x_2, t_2) d\Omega \quad (11)$$

Given snapshots of the i^{th} solution state vector $s_i(x, t) = \{s_i(x, t_1), \dots, s_i(x, t_n)\}$, s_i can either be the mesh deformation field q_i , the velocity field u_i or the pressure field p . Let $Y^{s_i} = \text{span}\{s_i(x, t_1), \dots, s_i(x, t_n)\}$, and $n^{s_i} = \dim(Y^{s_i})$. r^{s_i} is the reduced number of POD space modes (Φ^{s_i}) for s_i . Eigenmodes set obtained by solving the maximum likelihood estimation problem:

$$\max_{\psi^{s_i} \in H} \frac{(s_i, \psi^{s_i})^2}{\|\psi^{s_i}\|_H} = \max_{\Phi^{s_i}} \sum_{j=1}^{r^{s_i}} \frac{(s_i, \Phi^{s_i})^2}{(\Phi^{s_i}, \Phi^{s_i})} \quad (12)$$

where $\langle \cdot \rangle$ denotes a statistically average operator. The solution to Eq. (12) is provided by the set of left singular vectors $Y^{s_i}(Y^{s_i})^H$, and the POD space mode Φ^{s_i} can be obtained by performing singular value

decomposition on $Y^{s_i}(Y^{s_i})^H$:

$$(Y^{s_i}(Y^{s_i})^H - \lambda^{s_i} I) \Phi^{s_i} = 0 \quad (13)$$

where the $Y^{s_i}(Y^{s_i})^H$ is a $n^{s_i} \times n^{s_i}$ order real symmetric matrix. However, when the spatial dispersion is too large, especially when $r^{s_i} \ll n^{s_i}$, Eq. (13) is a large-scale eigenvalue problem, and it is inefficient to solve. In this regard, a snapshot POD method (Sirovich, 1987) is proposed to reduce the degree of freedom of the eigenvalue problem to r^{s_i} , that is, solve the following eigenvalue problem:

$$(Y^{s_i})^H Y^{s_i} \Phi^{s_i} - \Phi^{s_i} \Lambda^{s_i} = 0 \quad (14)$$

Then, the set of POD space modes Φ^{s_i} can be obtained. By using Reynolds decomposition, s_i can be represented by r^{s_i} POD modes:

$$s_i(x, t) \approx s_i(x) + s_i'(x, t, \theta) = s_i(x) + \sum_{j=1}^{r^{s_i}} a_j^{s_i}(t) \Phi_j^{s_i}(x) \quad (15)$$

where $a_j^{s_i}(t)$ are POD temporal coefficients for s_i , $s_i(x)$ represent the time average term, and $s_i'(x, t, \theta)$ represents the fluctuating part extracted as a linear combination of POD space modes $\Phi_j^{s_i}(x)$ and POD temporal coefficients $a_j^{s_i}(t)$ (Erwan and Hamdouni, 2010).

2.2. LSTM for POD-NIROM methodology

The selected POD modes have high energy content and are used to construct a low-dimensional subspace of the solution state vector $s(x, t)$. Substituting Eq. (14) into the bidirectional fluid-solid coupling equations and taking the POD space modes Φ^{s_i} as the shape functions, then weighted integrating the equations over the computational domain, the reduced order ordinary differential equations (ODEs) are obtained. The ODEs can be rewritten in the general form by Eq. (16):

$$F_j \left(\dot{a}_j^{q_i}, \ddot{a}_j^{q_i}, a_j^{q_i}, \dot{a}_j^{u_i}, \ddot{a}_j^{u_i}, a_j^{p_i}, t, \mathbf{d}_i^0 \Big|_{\Gamma_I} \right) = 0 \quad (16)$$

where $\dot{a}_j^{q_i}$ and $\ddot{a}_j^{q_i}$ represent the first and second derivatives of $a_j^{q_i}$ (POD temporal coefficients corresponding to the desired shape motion of the robot $\mathbf{d}_i^0 \Big|_{\Gamma_I}$ on Γ_I) with respect to time t , respectively. And $\dot{a}_j^{u_i}$ represents the first derivative of $a_j^{u_i}$ (POD temporal coefficients for fluid velocity u_i) with respect to time t .

As it performs well in nonlinear dimensionality reduction and sequential data learning, LSTM deep learning methods are used to construct the function \mathcal{F}_j . The function \mathcal{F}_j is used to approximate the function F_j in Eq. (16), where the POD temporal coefficients of the selected POD modes are determined. The LSTM neural network

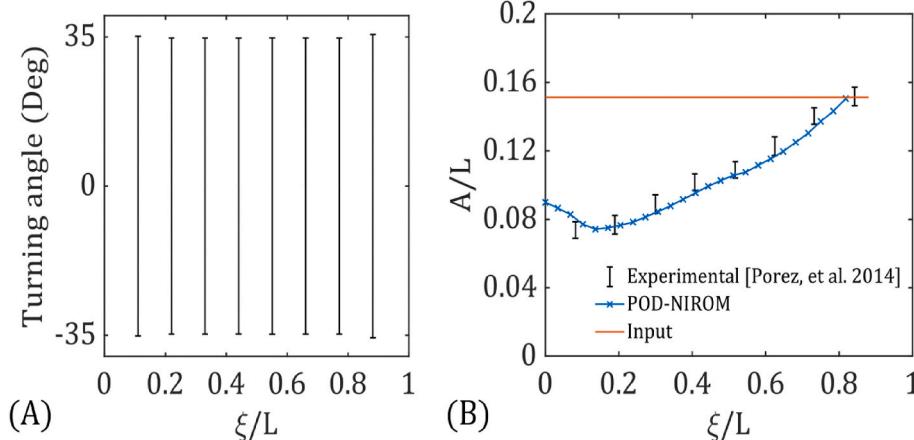


Fig. 2. (A) Joint oscillating amplitude that matches the coefficients ($A_0 = 0.1512$, $A_1 = 0$, and $A_2 = 0$). (B) The lateral displacement envelope of the robot. The red line indicates the input (the envelope of the desired shape motion of the robot), the black markers represent the experiment, and the blue line represents the envelope of robot undulation predicted by POD-NIROM (taking the red line as input). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

was proposed by Hochreiter in 1997 (Hochreiter and Schmidhuber, 1997). And the input gate i_t , the cell state c_t , the forget gate f_t , and the output gate o_t are the four components of the single LSTM neural network cell. The cell state c_t contains the input and the state preserved from the previous unit. The input gate i_t controls the transmission of the information from the input to the cell state at the current moment. The forget gate f_t controls the state preserved from the cell at the previous moment to the current moment. The output gate o_t controls the information sent to other nodes. The updated cell state \tilde{c}_t describes the output of the previous unit and the input of the current unit. The schematic of the LSTM Layer architecture and a single LSTM cell structure are shown in Fig. 3(B)(C).

In the LSTM neural network, the forward pass equations can be used to calculate the mapping from the input $x = (x^1, \dots, x^{N_t})$ to the output $h = (h^1, \dots, h^{N_t})$. The compact forms of the equations are given as follows:

$$f_t = \sigma \left([W_{fh} \quad W_{fx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_f \right) \quad (17)$$

$$i_t = \sigma \left([W_{ih} \quad W_{ix}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_i \right) \quad (18)$$

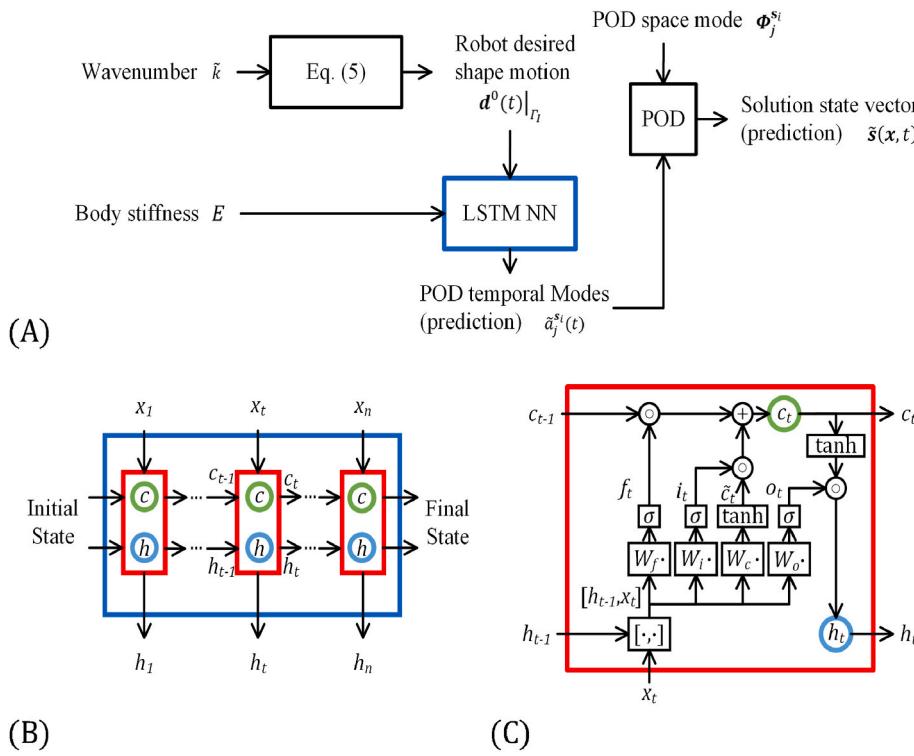
$$\tilde{c}_t = \tanh \left([W_{ch} \quad W_{cx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_c \right) \quad (19)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (20)$$

$$o_t = \sigma \left([W_{oh} \quad W_{ox}] \begin{bmatrix} h_{t-1} \\ c_t \end{bmatrix} + b_o \right) \quad (21)$$

$$h_t = o_t \circ \tanh(c_t) \quad (22)$$

where t indexes the time step, $x_t \in \mathbb{R}^x$ is the input vector to the LSTM unit, $h_t \in \mathbb{R}^h$ the hidden state vector (or called the output vector of the LSTM unit), $\tilde{c}_t \in \mathbb{R}^h$ is the updated cell state, and $c_t \in \mathbb{R}^c = \mathbb{R}^h$ is called cell state vector. $i_t \in \mathbb{R}^h$, $f_t \in \mathbb{R}^h$, and $o_t \in \mathbb{R}^h$ denote the input, forget,



and output gate's activation vectors, respectively. $b \in \mathbb{R}^h$ is the bias vector, W denotes the weight matrices (e.g. $W_f = [W_{fh} \quad W_{fx}] \in \mathbb{R}^{c \times (h+x)}$) is the weight matrix used to calculate the forget gate's activation vector f_t . σ is the sigmoid function, \circ is the Hadamard product, and the initial values $c_0 = 0$ and $h_0 = 0$.

The structure of the POD-NIROM used in this work is shown in Fig. 3 (A). In this work, the input of the LSTM neural network is the desired shape motion of the robot $d^0|_{T_1}$. And the outputs are the POD temporal coefficients predicted by the LSTM model $\tilde{a}_j^{st}(t)$.

The construction and prediction of NIROM can be summarized as follows:

- (1) Define a series of cases with various wavenumber $\tilde{k} = [\tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_{N_k}]$. Calculate the desired shape motion of the robot for each case $d^0(t; \tilde{k})|_{T_1}$.
- (2) Obtain the solution state vector for each case $s(x, t; \tilde{k})$ through the HF model simulation.
- (3) Obtain the POD bases $\Phi_j^{st}(x)$ by performing POD on the solution state vector $s_j(x, t; \tilde{k})$. And project the solution state vector onto each POD base to get the POD temporal coefficients $a_j^{st}(t; \tilde{k})$ for each case.
- (4) Generate the training cases with various wavenumber $\tilde{k}_{tr} = [\tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_{N_{tr}}]$. Use time-series $d_j^0(t; \tilde{k}_{tr})|_{T_1}$ as input data and time series $a_j^{st}(t; \tilde{k}_{tr})$ as output, train the LSTM neural network.
- (5) Generate the test cases with various wavenumber $\tilde{k}_{te} = [\tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_{N_{te}}]$. Treat the LSTM neural network as a function, $\mathcal{T}_j(a_j^{st}, d_j^0|_{T_1}) = 0$. Using the LSTM neural network, take the time series $d_j^0(t; \tilde{k}_{te})|_{T_1}$ as input, calculate the predicted POD temporal coefficients $\tilde{a}_j^{st}(t; \tilde{k}_{te})$ and compare it with the POD temporal

Fig. 3. Illustrate the proposed NIROM. (A) POD-NIROM Methodology. (B) The schematic of the LSTM NN architecture, the blue rectangle in Fig. 3(A). (C) The schematic of a single LSTM cell structure, the red rectangle in Fig. 3(B). First, the input is obtained according to the robot swimming kinematics. Then the LSTM NN is used to predict the POD temporal coefficients. Then, the predicted POD temporal coefficients are combined with the POD space modes to reconstruct the predicted solution state vector. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

- coefficients $a_j^{s_i}(t; \tilde{k}_{te})$ generated by the projection of the solution state vector $s_i(x, t; \tilde{k}_{te})$ on the POD bases $\Phi_j^{s_i}(x)$.
- (6) Use the predicted value of POD temporal coefficients $\tilde{a}_j^{s_i}(t; \tilde{k}_{te})$ and the POD bases $\Phi_j^{s_i}(x)$ to reconstruct the predicted value of the solution-state vector $\tilde{s}_j(x, t; \tilde{k}_{te})$ using Eq. (15).

2.3. Validation of the HF model

The mesh convergence analysis is performed, to obtain a reliable dynamic simulation. Three different meshing strategies were applied to the present HF model, precisely 12379 elements, 27707 elements, and 43299 elements. The current model is simulated using a mesh with 27707 elements since the simulation tends to converge. Fig. 4(E) depicts the time series of the forward swimming speed of the robotic fish (when $\tilde{k} = 1$).

In addition, multiple simulations show that, in the case with $\tilde{k} = 1$, when the body stiffness of the robot is $E = 0.7\text{MPa}$ and the Poisson's ratio is $\nu = 0.3$ (considered in Eq. (3) basing on the Generalized Hooke's Law), the simulation results agree well with the experimental measurements. Fig. 4(A)–4(D) presents the motion of the forward swimming robot relative to the fluid during a swimming cycle (between $3T$ and $4T$, when the time average hydrodynamic force acting on the robot is approximately zero). The experimental and the numerical simulation result of Porez et al. (2014) are used as a reference, of which the body axis are represented by red circles and blue dotted lines, respectively. At the same time, the black dashed line represents the contour line of the robot calculated by the present HF model. It can be seen that compared to the experimental results, the current HF model can provide a reliable data set for building the POD-NIROM.

3. Construction of POD-based non-intrusive ROMs

As depicted in Fig. 5(A), the forward swimming case was simulated for all 61 cases with various dimensionless wavenumber $\tilde{k}\epsilon[0.4, 1.6]$. For each case, snapshots are taken every 20 time-steps $dt = 20\Delta t = 0.05s$ during $te[0, 10]$ in seconds. Consequently, each set has a total of 201 snapshots, while each swing cycle covers 20 snapshots and the number of snapshots for all 61 cases is 12261. The Hold-out method is used to

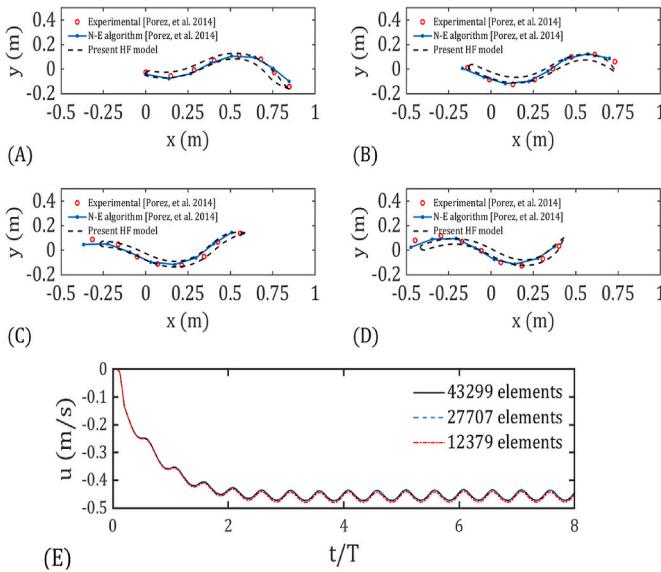


Fig. 4. The robot's straightforward swimming. (A) $t = 3T$, (B) $t = 3.25T$, (C) $t = 3.5T$, (D) $t = 4T$, (E) swimming speed at the center of the robot (from $t = 0T$ to $t = 8T$).

divide the data set. Considering the insufficient number of samples in the data set, to verify the stability, the partition was repeated 20 times, shown in Fig. 5(B).

The finite element error in the energy norm (L^2 norm) is used to estimate the error of reconstructing the data set using POD (Erwan et al., 2007; Sirovich, 1987; Haas and Zandbergen, 1996). The normalized reconstruction error in L^2 form ε^{s_i} is defined as:

$$\varepsilon^{s_i} = 1 - \frac{\sum_{i=1}^{N_r^{s_i}} \lambda_i^{s_i}}{\sum_{i=1}^{N^s_i} \lambda_i^s} \quad (23)$$

where $N_r^{s_i}$ is the number of POD modes used, which is much smaller than the total POD modes N^s_i . The reconstruction error of the energy captured by different amounts of POD modes is shown in Fig. 6. Consequently, to make POD-NIROM contain 99.8% of the energy, $N_r = [N_r^u, N_r^v, N_r^p, N_r^x, N_r^y] = [60, 60, 60, 10, 10]$ out of $N = [N^u, N^v, N^p, N^x, N^y] = [12261, 12261, 12261, 12261, 12261]$ are reserved for ROM, and the corresponding reconstruction error $\varepsilon = [\varepsilon^u, \varepsilon^v, \varepsilon^p, \varepsilon^x, \varepsilon^y] = [1.9 \times 10^{-2}, 1.1 \times 10^{-2}, 8.35 \times 10^{-6}, 1.38 \times 10^{-6}, 4.68 \times 10^{-4}]$.

The average value s_j and two POD space modes $\Phi_i^{s_i}$ ($i = 2, 5$) of the solution state vector $s_i(x, t)$ with $N = 12261$ snapshots are illustrated in Fig. 8.

Fig. 5(C) shows the verification error (loss function) of the LSTM neural network vs epochs. The verification method was repeated 20 times, and the training set and the test set are split as shown in Fig. 5(B). The half mean squared error (HMSE), used as the loss function for neural network training, is defined as follows,

$$\text{HMSE} = \frac{1}{2N_t} \sum_i^{N_t} \sum_j^{N_t} (\tilde{a}_{ij} - a_{ij})^2 \quad (24)$$

where N_t is the sequence length, which in this study is the number of snapshots for each case, $N_t = 201$, N_r is the number of responses, that is, the number of POD modes used, $N_r = 200$, \tilde{a}_{ij} is the network's prediction and a_{ij} is the target output. The loss function can be highly variable, and depends on whether the partition of the data set is sufficiently uniform. The average curve of the MSE indicates that after about 900 epochs, the validation will stabilize. Another metric for error estimation, is the coefficient of determination (R^2), which is defined as follows,

$$R^2 = 1 - \sum_i^{N_t} \sum_j^{N_t} (\tilde{a}_{ij} - a_{ij})^2 \Bigg/ \sum_i^{N_t} \sum_j^{N_t} (\hat{a}_{ij} - a_{ij})^2 \quad (25)$$

where \hat{a}_{ij} is the time average of the target POD temporal coefficients. R^2 is between zero and one. Fig. 5(D) depicts the R^2 obtained from the test set. It can be seen that the R^2 obtained for the test set does not appear unstable and the predicted POD temporal coefficients $\tilde{a}_j^{s_i}(t; \tilde{k}_{te})$ have been calculated with acceptable accuracy. The 9th partition with the best R^2 is adopted.

4. Results and discussion

The HF model simulation of 61 cases took 284025s in total. When building the PODNIROM model, the solution state vector imports and its POD analysis took 362s, and then training the neural network took 1343s. Finally, using PODNIROM to predict 61 cases, it took 156s in total. All processes are carried out on the AMD Ryzen Thread Ripper 1950X 16-Core Processor 3.40 GHz. It can be seen that POD-NIROM greatly reduces the calculation cost of robot propulsion parameter optimization. The response speed of POD-NIROM online prediction is 1820 times that of HF model simulation. The calculation cost of the whole process of POD-NIROM (including offline calculation and online prediction) is 0.6% of the HF model simulation.

The forward swimming process of the robot fish predicted by POD-

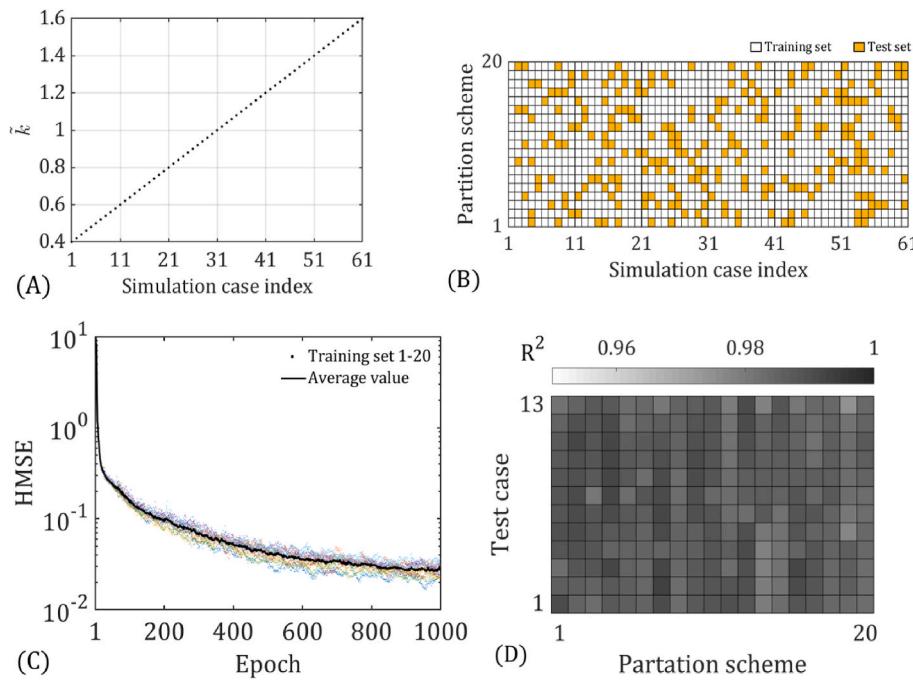


Fig. 5. Schematic of the parameters and training results of POD-NIROM. (A) The dimensionless wavenumber of each case. (B) 20 partition schemes of the training set and the test set. (C) Validation loss vs epochs for 20 partition schemes. (D) The coefficient of determination for 20 partition schemes.

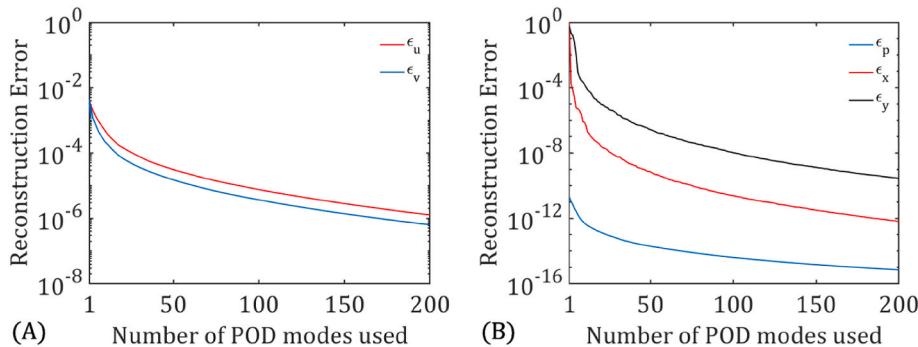


Fig. 6. The reconstruction error of the energy associated with the POD modes. (A) Velocity in the x-direction (red line), Velocity in the y-direction (blue line), pressure (black line). (B) Grid displacement in the x-direction (red line), grid displacement in the y-direction (black line). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

NIROM is compared with the experiment. The cases shown in Fig. 5(A) are all calculated when $\tilde{k} = 1.0$.

The input of POD-NIROM, that is, the time series of $d_n^0(t)$, at $\tilde{k} = 1.0$ is shown in Fig. 7. For convenience, only the nodes near the head of the robot ($[x, y] = [0.02\text{m}, 0.02\text{m}]$, node No. 110 node), the nodes near the tail ($[x, y] = [0.8737\text{m}, 0.0012\text{m}]$, No. 254 node), and the nodes near the center of gravity ($[x, y] = [0.442\text{m}, 0.02\text{m}]$, No. 171 node) are displayed.

Taking $d_n^0|_{\Gamma_I}$ at $\tilde{k} = 1.0$ as input, the prediction result of POD-NIROM is shown in Fig. 8(A). Each POD temporal coefficient has been de-normalized since the normalized data is used in the model. In this case, the present model gives a reliable prediction. In the selection of the rank of truncation N_r , both the reconstruction error ϵ and the frequency of the POD temporal coefficient are taken into consideration to avoid the distortion of the prediction result caused by the excessively high frequency of the POD temporal coefficient. The case where the prediction of the POD temporal coefficient is contaminated with higher frequencies has been discussed by Eivazi et al. (2020).

The POD result on the flow field (Fig. 8) shows that the first few energy-favored POD modes (taking the second and fifth POD spatial

bases as examples, $\Phi_i^{S_j}$ ($i = 2, 5$)) captured wake vortex features are mostly behind the tail within a body length. Considering that the POD spatial modes are ordered according to energy, that is, when moving forward, the swimmer's influence on the fluid is mainly concentrated near the body and within a body length range behind.

In this case, the predicted value of the solution state vector $\tilde{s}_j(x, t; k_{te})$ is reconstructed using Eq. (15). The performance of the flow characteristics captured by POD-NIROM is shown in Fig. 9(A)(B). Consistent with the swimmers that use body/caudal fin (BCF) undulations for propulsion (Bergmann and Iollo, 2011; Borazjani and Sotiropoulos, 2008), the wake generated by the robot (Fig. 9) has the characteristics of the thrust-indicative reversed Kaman vortex street (Jones and Platzer, 1997). Fig. 9(A)(B) also shows the velocity vectors and the vortex contour of the flow around the robot in the last prediction step ($t = 8T$), calculated by the HF model and predicted by the POD-NIROM, respectively. It is worth mentioning that the farther away from the robot in the backward direction, the fewer wake vortex features. This is caused by the selection of a small number of POD modes, meanwhile, the calculation cost is reduced.

The comparison of POD-NIROM results, HF model results, and

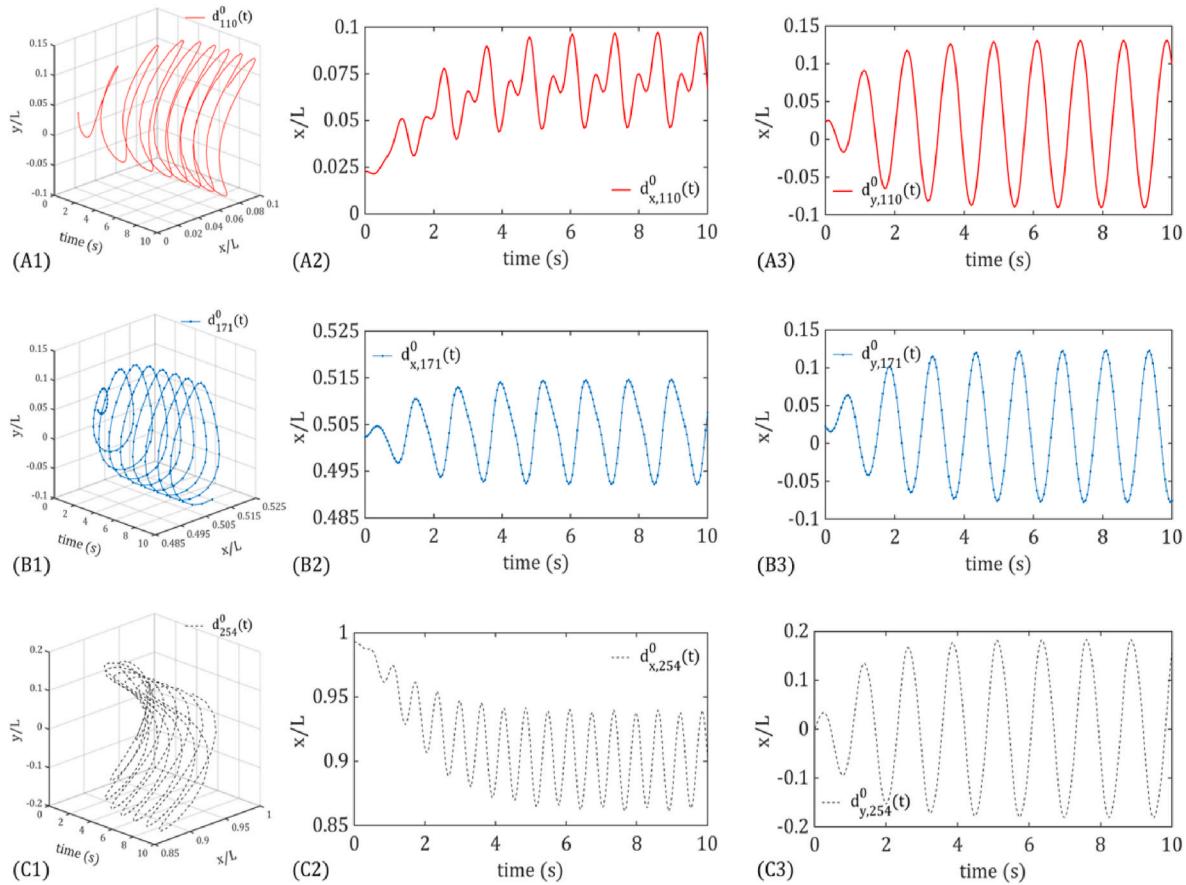


Fig. 7. Schematic of POD-NIROM input at $\tilde{k} = 1.0$. (A-) The robot head. (B-) The robot tail. (C-) The centroid of the robot. (-1) Three-dimensional schematic (one-dimensional time, two-dimensional space). (-2) Schematic in the x-direction. (-3) Schematic in the y-direction.

experimental results on a series of robot swimming snapshots sampled in a swimming cycle are displayed in Fig. 9(C)–9(F). In Fig. 9(C), the first snapshot, the robot's center of gravity obtained by the three methods is overlapped. At the same time, the robot head of the experimental result is taken as the zero, snapshots are captured every $0.25T$, a good match is observed. And the POD-NIROM with $[N_r^x, N_r^y] = [10, 10]$ can convincingly predict the deformation of the grid, especially the displacement on the fluid-solid coupling boundary, which is the movement of the robot.

To compare the lateral undulation amplitude of the robot predicted by the POD-NIROM with the experimental results more intuitively, the forward displacement of the robot is calculated. In a swimming cycle starting from $3T$, the forward displacement at the center of the robot is shown in Fig. 10(B). It can be seen that the prediction of POD-NIROM is consistent with the experiment.

In this case, with respect to the body length Lb , the time series of the velocity of forwarding swimming v' is presented in Fig. 10(A). It can be seen that after 4 swimming cycles ($4T$), the forward swimming speed of the robot tends to be stable. In the 4 swimming cycles from $t = 4T$ (from $4T$ to $8T$), the time average of the velocity ($v'_{HF} = -0.476 Lb/s$ and $v'_{POD} = -0.468 Lb/s$) are calculated. To illustrate the dependence of the swimming performance of the robot on the dimensionless wavenumber \tilde{k} , the variation of the robot's average forward swimming speed $v'(\tilde{k})$ (Average of the 8 swimming cycles, during $0 - 8T$) is plotted in Fig. 10(C). The prediction in Fig. 10(C) shows that the maximum time average velocity ($\max(v') = -0.58 Lb/s$) is reached, when $\tilde{k} = 0.75$. Compared with the experiment, the error of the simulation is composed as follows:

(i) to ensure the robot swims along the negative direction of the x-axis, the start up process (described in Section 2.1) is performed before starting to count swimming cycles (before $0T$, in other words, $t < 0.5s$);

(ii) the 2D assumption made judging from the domain dimensions is applied to simulate and predict the swimming performance of the robot;

(iii) for simplicity, the assumption of homogeneous material is used in the current simulations and predictions, but in the experiments, the robot is designed as a series of rigid links connected with soft joints.

In the HF model, the robot swims in the negative x direction, and the component of the instantaneous hydrodynamic force in the x-direction acting on the robot (F_{HF}) is calculated as follows:

$$F_{HF}(t) = \int_{\Gamma_I} (-pn_x + \tau_{xx}n_x + \tau_{xy}n_y) d\Gamma_I \quad (26)$$

$$\tau_{xx} = 2\mu u_x \quad (27)$$

$$\tau_{xy} = \mu u_y + \mu v_x \quad (28)$$

where $[n_x, n_y]^T$ is the unit normal vector on the fluid-structure coupling interface $d\Gamma_I$, τ_{ij} is the ij component of the viscous stress tensor. Substituting Eq. (15) into Eqs. (26–28), then F_{POD} is obtained:

$$F_{POD}(t) = \int_{\Gamma_I} \left(- \left(p + \sum_{j=1}^{r^p} a_j^p(t) \Phi_j^p \right) n_x + 2\mu \left(u + \sum_{j=1}^{r^p} a_j^u(t) (\Phi_j^u)_x \right) n_x \right. \\ \left. + \mu \left(u + \sum_{j=1}^{r^p} a_j^u(t) (\Phi_j^u)_y \right) n_y + \mu \left(v + \sum_{j=1}^{r^p} a_j^v(t) (\Phi_j^v)_x \right) n_y \right) d\Gamma_I \quad (29)$$

F_{HF} and F_{POD} are shown in Fig. 11(A). It can be found that after 4 swimming cycles ($4T$), the hydrodynamic force received by the robot tends to be stable, which is consistent with the findings from Fig. 11(A).

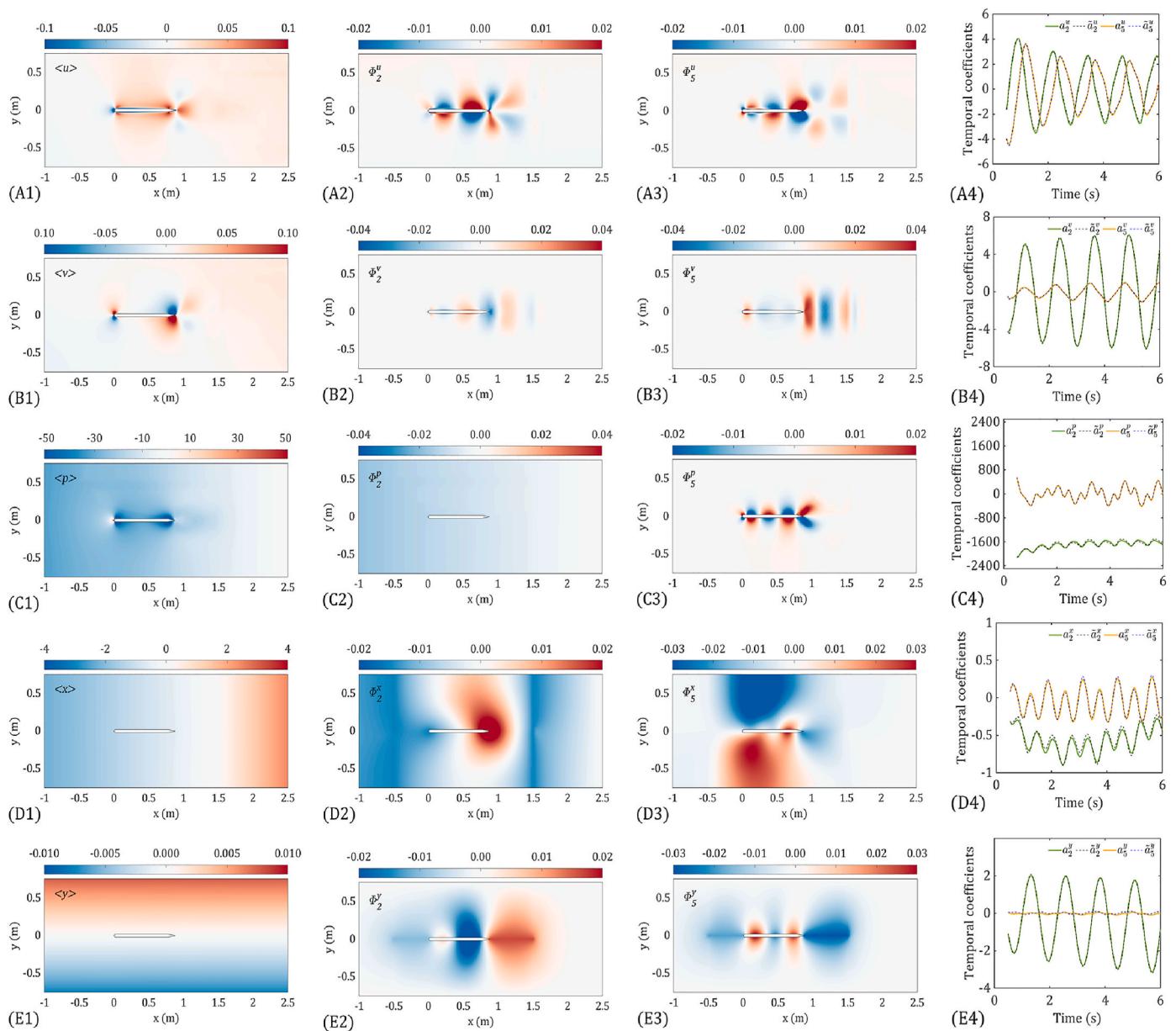


Fig. 8. Illustration of POD modes. (A-) POD space base: velocity in the x-direction. (B-) POD space base: velocity in the y-direction. (C-) POD space base: pressure. (D-) POD space base: grid displacement in the x-direction. (E-) POD space base: grid displacement in the y-direction. (-1) Time average. (-2) The second POD space base. (-3) The fifth POD space base. (-4) schematic of the POD temporal coefficient at $\tilde{k} = 1.0$. The data has been de-normalized. The solid line represents the reference data and the dotted line represents the prediction.

It should be noted that the robot is swimming along the negative direction of the x-axis, that is to say, when $F < 0$, the instantaneous hydrodynamic force provides thrust for the robot. Conversely, when $F > 0$, the instantaneous hydrodynamic force provides resistance for the robot. It can be found that in each swimming cycle, the instantaneous hydrodynamic force has 2 strokes (forward tail stroke and backward tail stroke), which is consistent with the experimental observations (Videler and Hess, 1984). Moreover, the prediction of POD-NIROM is in agreement with the HF simulation.

To illustrate the dependence of the swimming performance of the robot on the dimensionless wavenumber \tilde{k} , the variation of the robot's average thrust $Th(\tilde{k})$ (average value of 4 swimming cycles starting from $t = 4T$, during $4T - 8T$) is plotted in Fig. 11(B). Considering that when the swimming speed of the robot tends to be stable, the time average value of the hydrodynamic force in each swimming cycle tends to be 0.

For comparison, the hydrodynamic force $F_{HF}(t)$ is proposed to be decompose into thrust (the force that the robot receives in the same direction as the robot's forward motion) $Th_{HF}(t)$ and drag (the force on the robot in the opposite direction to the forward motion of the robot) $Dr_{HF}(t)$ (Borazjani and Sotiropoulos, 2008):

$$Th_{HF}(t) = \frac{1}{2}F_{HF}(t) + \frac{1}{2}\left(\left|\int_{\Gamma_I} -pn_x d\Gamma_I\right| + \left|\int_{\Gamma_I} (\tau_{xx}n_x + \tau_{xy}n_y) d\Gamma_I\right|\right) \quad (30)$$

$$Dr_{HF}(t) = -\frac{1}{2}F_{HF}(t) + \frac{1}{2}\left(\left|\int_{\Gamma_I} -pn_x d\Gamma_I\right| + \left|\int_{\Gamma_I} (\tau_{xx}n_x + \tau_{xy}n_y) d\Gamma_I\right|\right) \quad (31)$$

Substituting Eq. (15) into Eq. (30), then Th_{POD} is obtained:

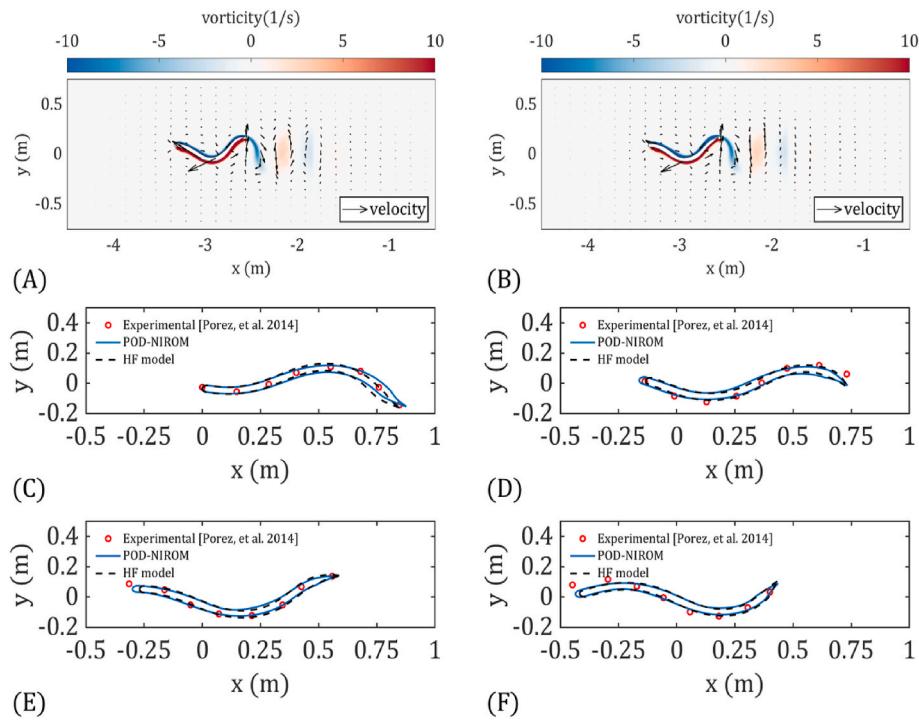


Fig. 9. Snapshot of the robot's straightforward swimming in a swimming cycle at $\tilde{k} = 1.0$. The schematics of the velocity vector and vortex contour at $t = 8T$ calculated by the HF model and the POD-NIROM are displayed in (A) and (B) respectively. (C)–(F) present the snapshots of the robot's straightforward swimming in a swimming cycle at $t = 3T$ (C), $t = 3.25T$ (D), $t = 3.5T$ (E), and $t = 4T$ (F).

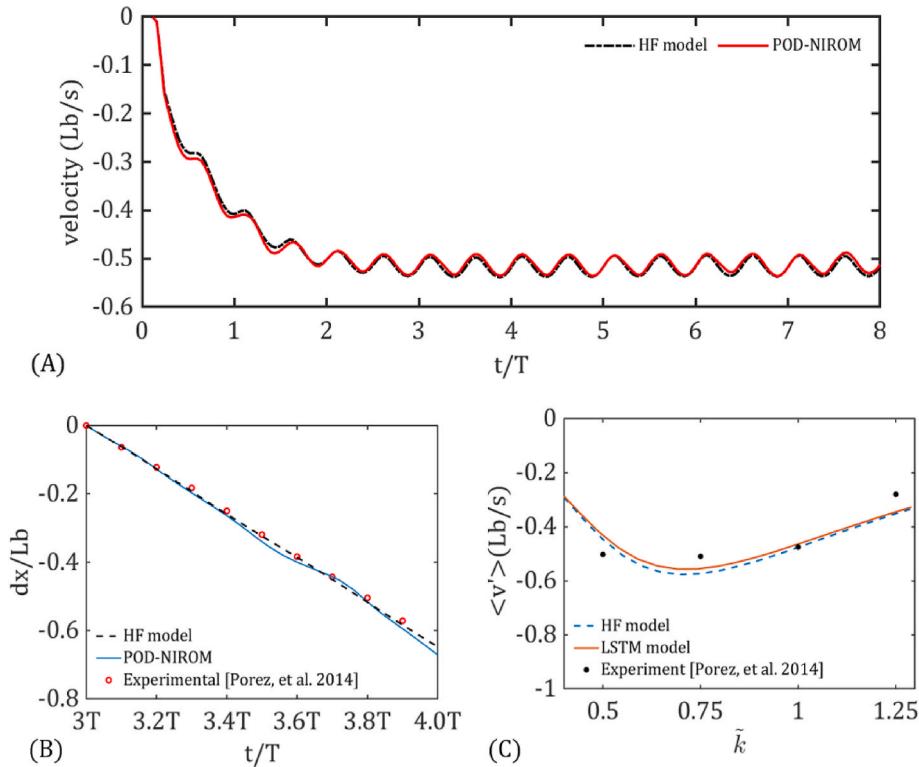


Fig. 10. Time history of the forward swimming performance at the center of the robot. (A) The forward swimming velocity ($\tilde{k} = 1.0$), (B) the forward displacement between $3T$ and $4T$ ($\tilde{k} = 1.0$), (C) the average forward swimming speed for varies \tilde{k} . It is worth mentioning that the robot swims along the negative direction of the x-axis.

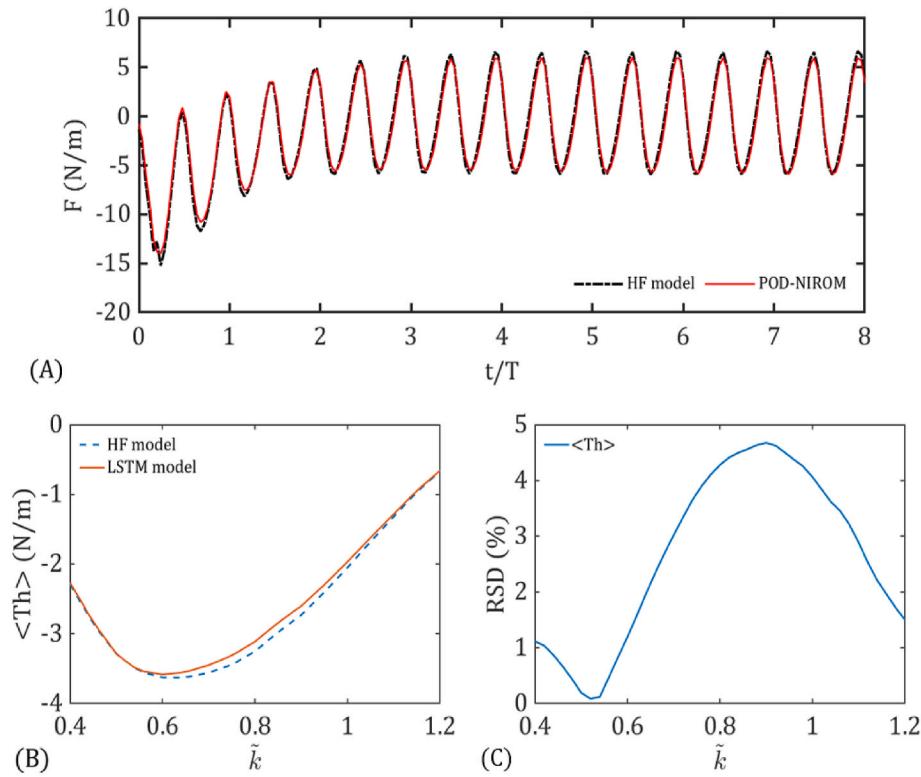


Fig. 11. The instantaneous force on the robot during forward swimming. (A) The instantaneous hydrodynamic force ($\tilde{k} = 1.0$), (B) the average thrust for varies \tilde{k} , (C) the error of the average thrust between the POD-NIROM prediction and the simulation of the HF model.

$$\begin{aligned} Th_{\text{POD}}(t) = & \frac{1}{2}F_{\text{POD}}(t) + \frac{1}{2}\left| \int_{\Gamma_I} - \left(p + \sum_{j=1}^{r^p} a_j^p(t) \Phi_j^p \right) n_x d\Gamma_I \right| \\ & + \frac{1}{2}\left| \int_{\Gamma_I} \left(2\mu \left(u + \sum_{j=1}^{r^u} a_j^u(t) (\Phi_j^u)_x \right) n_x + \mu \left(u + \sum_{j=1}^{r^u} a_j^u(t) (\Phi_j^u)_y \right) n_y \right. \right. \\ & \left. \left. + \mu \left(v + \sum_{j=1}^{r^v} a_j^v(t) (\Phi_j^v)_x \right) n_y \right) d\Gamma_I \right| \end{aligned} \quad (32)$$

On the one hand, as shown in Fig. 10(C), the maximum time average velocity ($\max(v') = -0.58 \text{ Lb/s}$) is reached when $\tilde{k} = 0.75$. Simultaneously, the peak of the time average propulsion force ($\max(Th) = -3.58 \text{ N/m}$) appears when $\tilde{k} = 0.62$ (see Fig. 11(B)).

On the other hand, as shown in Figs. 10(C), 61 POD-NIROM predictions and the corresponding 61 HF model's simulations are compared with 5 sets of experiments. The comparison shows that the prediction of POD-NIROM has a consistent trend with the experimental measurement. The body stiffness of the predicted robot does not match the experiment is the main reason for the inconsistency between the prediction and the simulation in Fig. 10(C), beyond the truncation error when constructing the POD-NIROM and the error in the experiment. Due to the lack of data, when defining the body stiffness of the robot, only the measurement when $\tilde{k} = 1$ is referred to. Therefore, the error between prediction and measurement when $\tilde{k} = 1$ is significantly smaller than in other cases.

Note that when comparing the POD-NIROM prediction with the HF model simulation, the difference in the time average propulsion force h (see Fig. 11(B)) is larger than the time average speed v' (see Fig. 10(C)). The calculation of v' uses some POD modes ($s_i, a_j^{s_i}$ and $\Phi_j^{s_i}$ when $s_i = x, y$), which mainly reflects the truncation error ϵ_x, ϵ_y and the prediction error of a^x and a^y . However, the equation of Th (see Eq. (30)) uses all the POD modes ($s_i, a_j^{s_i}$ and $\Phi_j^{s_i}$ when $s_i = u, v, p, x, y$). The difference between

Th 's prediction and simulation, indicating all truncation errors of the ROM and the prediction error of all POD temporal coefficients, is described by the relative standard deviation (RSD). Fig. 11 (C) describes the RSD of Th , which shows the maximum $\max(\text{RSD}) = 4.67\%$ and the average $\overline{\text{RSD}} = 2.60\%$. The results show that the proposed POD-NIROM can predict quickly and reliably for cases where \tilde{k} is within the range covered by the training set ($\tilde{k} \in [0.4, 1.6]$).

5. Conclusion

In this paper, an effective model (POD-NIROM) is proposed, which makes full use of Long short-term memory Neural Network (LSTM NN) and proper orthogonal decomposition (POD) to predict the solution state vector of the two-way fluid-structure interaction problem. And the proposed POD-NIROM takes into account the body stiffness (described by elastic modulus), which is an important internal parameter of the robot when predicting the underwater swimming performance of the robot. The predictions of the proposed POD-NIROM are more valuable than the model that regard the robot swimming underwater as a rigid body motion. This is the first time that the proposed model has been used to optimize the propulsion parameters of an underwater vehicle, in which the body stiffness of the robot has been taken into consideration.

Compared with the HF model, the proposed POD-NIROM has a lower computational cost when a large number of simulations (for robot parameter optimization), or fast predictions (for robot propulsion control) are required. Based on the POD method, 200 groups of POD modes (extracted from 61305 groups) are extracted from the solution state vector for the construction of the reduced-order model. To reduce the difficulty of constructing the proposed model, the deep learning algorithm is introduced to predict the evolution of POD time coefficients (instead of the Galerkin projection scheme). The prediction and simulation of robot swimming performance are compared with experimental measurements. The results show that the lateral undulation and forward

swimming of the robot in the prediction and simulation have a consistent trend with the experiment.

The time average propulsion is also presented. The maximum value of the relative standard deviation (RSD) between the POD NIROM prediction and the HF model simulation is 4.67%, and the average value is 2.60%. Simultaneously, the dimensionless wavenumber of the robot body used for propelling is optimized. It is found that for various dimensionless wavenumber, the time-averaged propulsion force and the time-averaged speed have a consistent trend. And there is a dimensionless wavenumber that makes the robot's average thrust and robot average forward swimming speed reach the maximum, at the same time.

These results prove that the prediction of the proposed POD-NIROM is reliable and fast, which highlights the generalization ability of deep learning neural networks and is of great significance to the research of the optimization of bio-inspired robots.

Data availability

The code and parameter settings of the POD-NIROM proposed in this paper are available through the link: <https://github.com/AX1ANG/PODNIROM>.

CRediT authorship contribution statement

Zixiang Ying: Conceptualization, Data curation, Formal analysis, Investigation, Roles/Writing, Writing – original draft, Visualization. **Linxiang Wang:** Methodology, Funding acquisition, Supervision, Validation. **Roderick Melnik:** Methodology, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 51575478 and Grant No. 61571007), China.

References

- Bao, Y., et al., 2011. Numerical prediction of aerodynamic characteristics of prismatic cylinder by finite element method with Spalart–Allmaras turbulence model. *Comput. Struct.* 89 (3), 325–338.
- Bergmann, M., Iollo, A., 2011. Modeling and simulation of fish-like swimming. *J. Comput. Phys.* 230, 329–348.
- Borazjani, I., Sotiropoulos, F., 2008. Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes. *J. Exp. Biol.* 211, 1541–1558.
- Borazjani, I., Sotiropoulos, F., 2009. Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *J. Exp. Biol.* 212, 576–592.
- Brooks, A., Hughes, T., 1982. Streamline upwind/petrov-galerkin Formulations for convection dominated Flows with particular Emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.* 32, 199–259.
- Choi, H.G., Yoo, J., 1997. A fractional four-step finite element formulation of the unsteady incompressible Navier-Stokes equations using SUPG and linear equal-order element methods. *Comput. Methods Appl. Mech. Eng.* 143, 333–348.
- Chorin, A., 1967. A numerical method for solving incompressible viscous flow problems*. *J. Comput. Phys.* 135, 12–26.
- Christianson, C., et al., 2019. Jellyfish-inspired soft robot Driven by fluid electrode dielectric organic robotic actuators. *Front. Robot. AI* 6, 126.
- Deufhard, P., 2006. Newton methods for nonlinear problems. *Affine invariance. Adapt. algorithm.* 35.
- Dragomirescu, I., Resiga, R., Muntean, S., 2013. Proper Orthogonal Decomposition Method in Swirling Flows Appl. 1558, 1349–1352.
- Eivazi, H., et al., 2020. Deep Neural Networks for Nonlinear Model Order Reduction of Unsteady Flows. *Phys. Fluids.* 32, 104–105.
- Erwan, L., Hamdouni, A., 2010. Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder. *J. Fluid Struct.* 26, 292–311.
- Erwan, L., Benouicha, M., Hamdouni, A., 2007. Proper orthogonal decomposition investigation in fluid structure interaction. *Eur. J. Comput. Mech. Européenne de Mécanique Numérique* 16.
- Fang, F., et al., 2013. Non-linear Petrov–Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. *J. Comput. Phys.* 234, 540–559.
- Farhat, C., et al., 1997. High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: Methodology and preliminary results. *Adv. Eng. Software* 28, 43–61.
- Gray, J., 1953. The locomotion of fishes. *Essays. Mar. Biol.* 1–16.
- Gray, J., 2020. Studies in animal locomotion. I. The movement of fish with special reference to the eel. *J. Exp. Biol.* 10, 88–104.
- Haas, P.C.A., Zandbergen, P.J., 1996. The Application of domain Decomposition to time-domain Computations of nonlinear water Waves with a panel method. *J. Comput. Phys.* 129, 332–344.
- Halder, R., Damodaran, M., Khoo, B., 2020. Deep learning based reduced order model for airfoil-gust and aeroelastic interaction. *AIAA J.* 58, 1–18.
- Hecht, F., 2012. New development in FreeFem++. *J. Numer. Math.* 20.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Jones, K., Platzter, M., 1997. Numerical computation of flapping-wing propulsion and power extraction. *AIAA 97*, 97–826.
- Katzschmann, R., Marchese, A., Rus, D., 2014. Hydraulic Autonomous Soft Robotic Fish for 3D Swimming. *Exp. Robot* 109, 405–420.
- Kherad, M., Moayyedi, M.K., Fotouhi, F., 2020. Reduced order framework for convection dominant and pure diffusive problems based on combination of deep LSTM and POD/DMD methods. *Int. J. Numer. Methods Fluid.* 93, 853–873.
- Kim, P., 2017. Deep Learning, in MATLAB Deep Learning: with Machine Learning, Neural Networks And Artificial Intelligence. In: Kim, P. (Ed.). *Apress, Berkeley, CA*, pp. 103–120.
- Lee, K., Kou, J., Zhang, W., 2019. Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. *Nonlinear Dynam.* 96, 1–21.
- Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* 807, 155–166.
- Liu, J., Hu, H., 2010. Biological inspiration: from carangiform Fish to multi-joint robotic fish. *JBE* 7, 35–48.
- Long, J.J., et al., 2011a. Inspired by sharks: a biomimetic skeleton for the flapping, propulsive tail of an aquatic robot. *Mar. Technol. Soc. J.* 45, 119–129.
- Long, J.J., et al., 2011b. Testing biomimetic structures in bioinspired robots: how vertebrae control the stiffness of the body and the behavior of fish-like swimmers. *Integr. Comp. Biol.* 51, 158–175.
- Lozovskiy, A., et al., 2016. POD-based model reduction for stabilized finite element approximations of shallow water flows. *J. Comput. Appl. Math.* 302, 50–70.
- Lumley, J., 1967. The structure of inhomogeneous turbulence. *Atmos. Turbul. Radio Wave Propag* 790, 166–178.
- Marchese, A., Onal, C.D., Rus, D., 2014. Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. *Soft Robot.* 1, 75–87.
- Masud, A., Hughes, T., 1997. A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. *Comput. Methods Appl. Mech. Eng.* 146, 91–126.
- Masud, A., Bhanabhagvanwala, M., Khurram, R.A., 2007. An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction. *Comput. Fluids* 36 (1), 77–91.
- Mohamed, K., et al., 2021. A Stress Mapping Immersed Boundary Method for Viscous Flows. <https://doi.org/10.37934/arfnts.87.3.120>, 87: p. 1-20.
- Mohan, A., Gaitonde, D., 2018. A Deep Learning Based Approach to Reduced Order Modeling for Turbulent Flow Control Using LSTM Neural Networks.
- Necas, J., Hlavacek, I., 1981. Mathematical Theory of Elastic and Elasto-Plastic Bodies: An Introduction. <https://doi.org/10.1016/c2009-0-12554-0>, 3.
- Nguyen, P.L., Do, V.P., Lee, B.R., 2012. Dynamic modeling of a flexible tail fin for a robotic fish. In: ICMT 2012-16th International Conference on Mechatronics Technology, pp. 431–436.
- Perrin, R., et al., 2006. 3D circular cylinder. https://doi.org/10.1007/978-3-540-39507-2_36, 299–312.
- Persillon, H., Braza, M., 1998. Physical analysis of the transition to turbulence in the wake of a cylinder by three-dimensional Navier-Stokes simulation. *J. Fluid Mech.* 365, 23–88.
- Phamduy, P., et al., 2016. Miniature Underwater Robotic Fish for Animal-Robot Interactions. *V002T17A009*.
- Placzek, A., Sigrist, J.-F., Hamdouni, A., 2009. Numerical Simulation of an oscillating Cylinder in a cross-Flow at low Reynolds number : Forced and free oscillations. *Comput. Fluids/Comput. Fluids* 38.
- Porez, M., Boyer, F., Ijspeert, A.J., 2014. Improved Lighthill fish swimming model for bio-inspired robots: modeling, computational aspects and experimental comparisons. *Int. J. Robot. Res.* 33, 1322–1341.
- Răzvan, Ș., Sandu, A., Navon, I.M., 2014. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *Int. J. Numer. Methods Fluid.* 76 (8), 497–521.
- Rabault, J., et al., 2019. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* 865, 281–302.
- Reichert, R., et al., 1994. Proper orthogonal decomposition applied to turbulence in a square duct. *Physics of Fluids - PHYS FLUIDS* 6, 3086–3092.

- Renganathan, S.A., Maulik, R., Rao, V., 2020. Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil. *Phys. Fluids* 32.
- Schilders, W.H.A., Vorst, H.A.V.D., Rommes, J., 2008. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, Berlin Heidelberg.
- Schlegel, et al., 2015. On long-term boundedness of Galerkin models. *J. Fluid Mech.* 765.
- Shen, Q., et al., 2013. Hydrodynamic performance of a biomimetic robotic swimmer actuated by ionic polymer-metal composite. *Smart Mater. Struct.* 22, 075035.
- Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. I - Coherent structures. II - Symmetries and transformations. III - Dynamics and scaling. *Q. Appl. Math.* 45.
- Srinivasan, P., et al., 2019. Predictions of turbulent shear flows using deep neural networks, 4, pp. 054–603. <https://doi.org/10.1103/physrevfluids.4.054603>. Issue: 5.
- Stankiewicz, W., et al., 2016. Modal decomposition-based global stability analysis for reduced order modeling of 2D and 3D wake flows. *Int. J. Numer. Methods Fluid.* 81 (3), 178–191.
- Storti, B., et al., 2019. Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach. *Energy* 186, 115–814.
- Triantafyllou, M.S., Triantafyllou, G.S., 1995. An efficient swimming machine. *Sci. Am.* 272 (3), 40–45, 48.
- Tu, J., et al., 2013. On dynamic mode decomposition: theory and applications. *J. Comput. Dynam.* 1.
- Tytell, E.D., et al., 2010. Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming. *Proc. Natl. Acad. Sci. U. S. A.* 107 (46), 19832–19837.
- Videler, J., Hess, F., 1984. Fast continuous swimming of two pelagic predators, saithe (*Pollachius virens*) and mackerel (*Scomber scombrus*): a kinematic analysis. *J. Exp. Biol.* 109, 209.
- Videler, J.J., Wardle, C.S., 1991. Fish swimming stride by stride: speed limits and endurance. *Rev. Fish Biol. Fish.* 1 (1), 23–40.
- Wang, H., et al., 2014. Wake-induced vibrations of an elastically mounted cylinder located downstream of a stationary larger cylinder at low Reynolds numbers. *J. Fluid Struct.* 50.
- Webb, P., 1984. Body form, Locomotion and Foraging in aquatic vertebrates. *Integr. Comp. Biol.* 24.
- Wu, T., 1961. Swimming of a waving plate. *J. Fluid Mech.* 10.
- Xiao, D., et al., 2013. Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair. *Comput. Methods Appl. Mech. Eng.* 255 (MAR.1), 147–157.
- Yu, J., et al., 2004. Development of a biomimetic robotic Fish and its control algorithm. *IEEE transactions on systems, man, and cybernetics. Part B, cybernetics : a publication of the IEEE systems, Man, Cybern. Soc.* 34, 1798–1810.
- Z, et al., 2016. Non-intrusive reduced order modelling with least squares fitting on a sparse grid. *Int. J. Numer. Methods Fluid.* 83 (3), 291–306.
- Zhao, M., et al., 2007. Hydrodynamic forces on dual cylinders of different diameters in steady currents. *J. Fluid. struct.* 23, 59–83.
- Zhao, Y., et al., 2019. A modified Proper Orthogonal Decomposition method for flow dynamic analysis. *Comput. Fluids* 182.
- Zhao, X., et al., 2020. Reduced order modeling-based discrete unified gas kinetic scheme for rarefied gas flows. *Phys. Fluids* 32, 67108.