Dynamic Modeling of Robotic Fish considering Background Flow using Koopman Operators

Xiaozhu Lin, *Graduate Student Member, IEEE*, Song Liu, *Member, IEEE*,
Chengyuan Liu, and Yang Wang, *Member, IEEE*

***Abstract*— Dynamic model is essential for robust and reliable robotic fish motion control. Despite considerable efforts in robotic fish dynamic modeling, background flow has not been well considered yet, leading to the deterioration of applying robotic fish to practice. In this paper, we propose a novel dynamic model, termed Flow-Aware Robotic fish Model (FARM), that with well consideration to background flow using Koopman operators without increasing computation complexity. Specifically, we first collect motion data of the robotic fish in different background flow fields, and then obtain a linear approximation (the dynamic model) of nonlinear dynamics through carefully selected lifted functions. The obtained model can predict the next state based on the current state, control input, and average flow velocity of the local flow field. We evaluate the effectiveness of obtained model by comparing the Root Mean Square Error (RMSE) of predicted motion trajectories with real trajectories in various flow field environments. The results indicate that FARM is highly promising for obtaining a reliable dynamic model and can achieve comparable prediction accuracy even in unseen flow field environments with rough flow maps.**

I. Introduction

Robotic fish has gained substantial attention in robotics field [1]–[3] in past decades due to their unique benefits, such as concealment, flexibility, and energy efficiency. Notable advancements have been achieved in various aspects, including the realm of electromechanical design [4], [5], underwater perception [6], [7] and control [8], [9]. However, the problem of dynamic modeling for the robotic fish, as playing a crucial role in motion planning and control, remains unsolved and challenging due to its inherently more complex high-dimensional and nonlinear properties compared to traditional Automatic Underwater Vehicle (AUV), especially in real-world environments with background flow.

When it comes to dynamic modeling like for robotic fish, the fundamental approaches are to derive them from first principles [10], [11]. However, such solutions usually undergo significant simplification, hence demanding complex expert knowledge and may lead to an inaccurate model. Furthermore, to obtain usable models, a large amount of real-world data to identify the unknown parameter [10] is still mandatory. Inspired by this fact, several works [3], [12] have circumvented any a prior knowledge of first principles and, instead, directly employ real-world data to train a Neural Network (NN) to describe the dynamic relationships of the desired states relevant to robotic fish. This eliminates the complexities and inaccuracies brought by approximation operation, enabling the direct acquisition of an NN-based dynamic model. However, the dynamic model based on the black box is not friendly for many control techniques. Alternatively, Mamakoukas *et al.* [13] and Castano *et al.* [14] used the Koopman operator [15], [16] to obtain the dynamic model of robotic fish, as it can transform a complex nonlinear system into a tractable linear system and the obtained model is explainable and control-friendly.

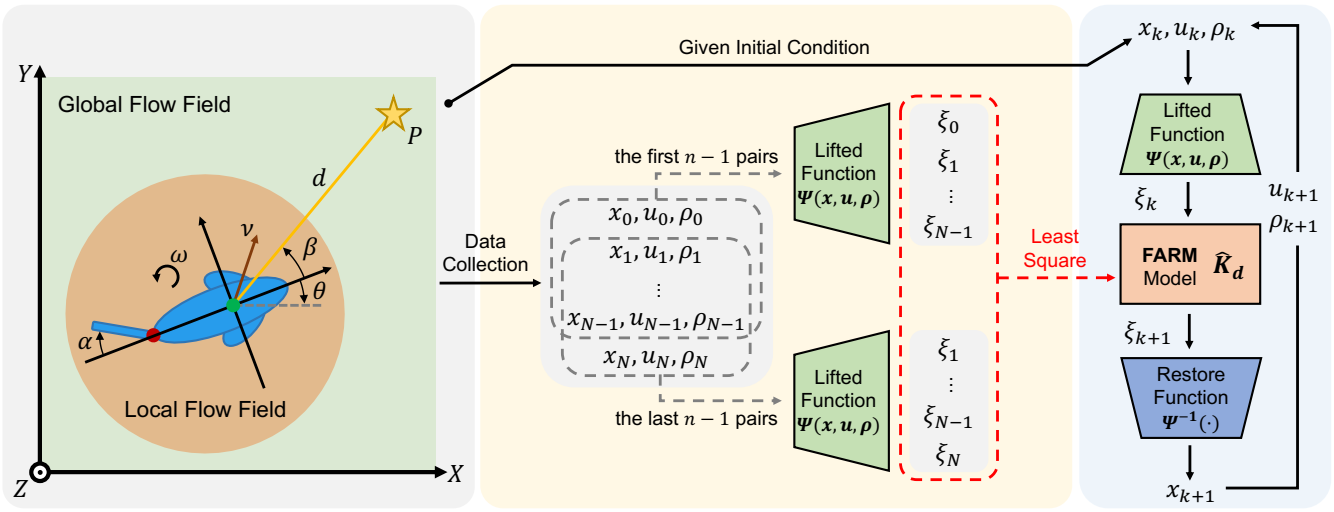
While the aforementioned studies have made substantial contributions to the dynamic modeling of robotic fish, they share a common limitation: the influence of background flow is not well considered during dynamic modeling, which leads that the established dynamic models are likely to fail to describe accurately when applied complicated environments with background flow, such as in ocean and river. Given the complex nature of interactive dynamic of the robotic fish in the background flow, researchers are often compelled to resort to end-to-end learning of intricate flow fields [17], [18]. These solutions circumvent the modeling problem of robotic fish in complicated environments, but the reliability and generality of the obtained control policy are limited. In addition, a notable fact that is numerous bio-mimetic sensors and method [19], such as the Artificial Lateral Line System (ALLS) system, have begun to emerge for sensing the flow field surrounding robotic fish, which undoubtedly expedites the urgency of considering the impact of background flow information on the dynamics of robotic fish as a reasonable model.

In this work, a novel dynamic modeling framework, named Flow-Aware Robotic fish Modeling (FARM), is proposed. Specifically, we firstly collect motion data of the robotic fish in a real platform with different background flows, where the velocity field of these background flows are known. Then, we divide the organized data into two groups, with a time step difference before and after, and use carefully selected lifted functions to expand the dimensions of the state. Finally, we obtain the linear approximation operator as dynamic model of the robotic fish using the least squares method. The effectiveness of obtained model is validated through extensive prediction experiments and comparison to the method without consideration to background flow information. The experimental results indicate that, even in unseen background flow scenarios, obtained model also has

This work was supported by the National Natural Science Foundation of China under Grant 62303321. *(Corresponding author: Yang Wang)*

Xiaozhu Lin, Song Liu, and Yang Wang are with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. *(E-mail: {linxzh, liuxp, wangyang4}@shanghaitech.edu.cn)*

Chengyuan Liu is with Aeronautical and Automotive Engineering, Loughborough University, Loughborough, UK LE11 3TU. *(E-mail: c.liu10@lboro.ac.uk)*



(a) (b) (c)

Fig. 1. **Overview of Pipeline.** (a) Schematic diagram of robotic fish undergoing planar locomotion with background flow. (b) Training the Koopman operator using customized lifted function with collected real world motion data. (c) Prediction of next state using trained Koopman model given current state, action and velocity of local background flow.

good prediction accuracy. In summary, FARM is a universal modeling framework with high accuracy, good robustness, and minimal data requirements for robotic fish without requirement of high accuracy in background flow information.

II. Flow-Aware Robotic Fish Model (FARM)

Robotic fish have complex geometric shapes and generate forward thrust through swinging motion. Compared to traditional propeller propelled AUVs, robotic fish have more sophisticated high-dimensional nonlinear dynamics, making it more challenging to establish dynamic model for robotic fish. The majority of existing works [10], [11], [13], [14] involve making several basic assumptions about hydrodynamics to obtain the model with undetermined parameters, and then identified these parameters by collecting motion data of the robotic fish in a static tank. Usually, the dynamic model $F(\cdot)$ is given in the form of

$$x_{k+1} = F(x_k, u_k) \tag{1}$$

where x stands for the state of robotic fish (e.g., position, attitude and velocity), while u is usually the control signal of joints of robotic fish.

However, the Eq. (1) possesses a critical limitation in that it is only applicable in scenarios where the robotic fish swim in an environment without any background flow. It is very intuitive that once the environment has a background flow, robotic fish will be subjected to flow-induced moment [19], the model will no longer be applicable, which also be proved in Section IV. To this end, we propose the Flow-Aware Robotic fish Model (FARM), which aims to incorporate the influence of background flow on the dynamic model of robotic fish. Although the flow field environment is exceptionally complex, considering that the Reynolds number of the environment where the robotic fish is located is less than 10^5 [20], we have simplified the model based on the following two important observations:

1) The motion state of robotic fish is only related to the local flow field around it.
2) The velocity of mean flow of the local flow field plays a dominant role in the dynamics of the robotic fish.

Therefore, the FARM model is as

$$x_{k+1} = F(x_k, u_k, \rho_k). \tag{2}$$

where ρ_k is represent the mean velocity of local flow field around the robotic fish.

In order to streamline the complexities inherent in this study and to better elucidate our central concepts, only planar locomotion of robotic fish is considered in this paper. As shown in Fig. 1a, the position of the robotic fish $p = [p_x, p_y]$ is represented by the Center of Mass (CoM) of the robotic fish. The head orientation and linear velocity of robotic fish expressed in inertial coordinates are denoted by θ and $v = [v_x, v_y]$, respectively. We use ω and α to denote the angular velocity and the tail deflection angle of robotic fish. Like most existing works [11], [14], the tail deflection angles α of robotic fish is controlled by the periodic function as

$$\alpha(t) = \alpha_b + \alpha_a \sin(\alpha_f * t) \tag{3}$$

where α_a, α_b, α_f are denoted as amplitude, bias, and frequency of the sinusoidal signal, respectively. By changing these quantities over time, the movement mode of the robotic fish will also change accordingly. Based on above statement, the specific content of each variable in Eq. (2) follows as

$$\begin{aligned} x_k &= [p_x, p_y, \theta, v_x, v_y, \omega]^{\mathrm{T}} \\ u_k &= [\alpha_a, \alpha_b, \alpha_f]^{\mathrm{T}} \\ \rho_k &= [\rho_x, \rho_y]^{\mathrm{T}} \end{aligned} \tag{4}$$

It is worth mentioning that the core idea of this work is to consider the flow field information in the form of the average velocity of the local flow field into the dynamic model of the robotic fish, in order to improve the motion ability of

the robotic fish in non stationary flow field environments. The way to obtain velocity field information within the range of motion is not the focus of this work, and there have been corresponding works focused on this field, such as motion tomography [19]. Another key point is that we hope that the dynamic model obtained through data-driven methods can be used for controller design. Therefore, we use the Koopman operator [16] to obtain linear approximations of complex nonlinear dynamics Eq. (2) in this work. The resulting linear system not only has good fitting performance, but also facilitates controller design.

III. Modeling using Koopman Operator

As a method for evolving complex nonlinear systems in a linear manner, Koopman operator enables controllers designed based on it to have better control performance than those that rely on approximating nonlinear systems near equilibrium points, which has attracted a lot of attention in the field of robotics, especially for soft robotics [21].

A. Finite-dimensional Koopman Operator Approximation

The Koopman operator is a linear infinite dimensional operator which can transform the evolution of nonlinear finite dimensional system in a linear but infinite dimensional way. In practice, data-driven methods, such as Extended Dynamic Mode Decomposition (EDMD) [15], are usually used to obtain manageable finite dimensional approximation.

Given a discrete-time nonlinear dynamical system with control input evolving as Eq. (2), where $x_k \in \mathbb{R}^6$, $u_k \in \mathbb{R}^3$ and $\rho_k \in \mathbb{R}^2$ in this paper. The Koopman operator is an infinite-dimensional operator that can express a finite-dimensional nonlinear system as a linear one, which is defined as

$$\begin{aligned}\Psi(x_{k+1}, u_{k+1}, \rho_{k+1}) &= \Psi(F(x_k, u_k, \rho_k), u_{k+1}, \rho_{k+1}) \\ &= \mathcal{K}\Psi(x_k, u_k, \rho_k)\end{aligned} \tag{5}$$

where $\Psi(\cdot)$ is the lifted function (i.e., observation function). Therefore, the Koopman operator $\mathcal{K}$ propagates the lifted state forward as the original dynamics $F(\cdot)$ do, but linearly.

In order to obtain the tractable finite-dimensional operator, the lifted function $\Psi(\cdot)$ is defined as vector-valued function as

$$\begin{aligned}\xi &= \Psi(x, u, \rho) \\ &= [x^{\mathrm{T}}, u^{\mathrm{T}}, \rho^{\mathrm{T}}, \psi_1(x, u, \rho), \cdots, \psi_n(x, u, \rho)]\end{aligned} \tag{6}$$

where $\Psi(\cdot) \in \mathbb{R}^N$ and $\xi \in \mathbb{R}^N$ is lifted observables. It should be noted that we can now obtain the original state x by selecting the first n term of the lifted state. Furthermore, we denote this restore operation as $\Psi^{-1}(\cdot)$[1], i.e.,

$$\Psi^{-1}(\xi_k) = \xi_k \begin{bmatrix} I^{6\times 6} \\ 0 \end{bmatrix} = x_k \tag{7}$$

Once we have collected the interaction trajectories, whether the simulator or the real world, between the agent and the system as

$$\begin{aligned}X &= [x_1, \cdots, x_P] \\ U &= [u_1, \cdots, u_P] \\ \Omega &= [\rho_1, \cdots, \rho_P]\end{aligned} \tag{8}$$

where $X \in \mathbb{R}^P$, $U \in \mathbb{R}^P$ and $\Omega \in \mathbb{R}^P$ are the sequence of the state, control input, velocity of local flow field, respectively. In general, the time gap Δt between x_k and x_{k+1} should be equally.

The approximate Koopman operator can be obtained by using the least squares method to fit the collected data as

$$\hat{\mathcal{K}}_d = \arg\min_{\hat{\mathcal{K}}_d} \sum_{k=1}^{P-1} \frac{1}{2} ||\xi_{k+1} - \hat{\mathcal{K}}_d \xi_k||^2 \tag{9}$$

where $\hat{\mathcal{K}}_d \in \mathbb{R}^{N\times N}$ is the approximate Koopman operator.

Since the parameter matrix i.e., approximate Koopman operator $\hat{\mathcal{K}}_d$ is linear, there is a closed-form solution for this linear least squares problem as

$$\hat{\mathcal{K}}_d = \mathcal{A}\mathcal{B}^{\dagger} \tag{10}$$

where † denotes the Moore-Penrose pseudo-inverse, and

$$\mathcal{A} = \frac{1}{P}\sum_{k=1}^{P-1} \xi_{k+1}\xi_k^{\mathrm{T}} \quad \mathcal{B} = \frac{1}{P}\sum_{k=1}^{P-1} \xi_k \xi_k^{\mathrm{T}} \tag{11}$$

Once the FARM model obtained, we can use it for state prediction. Given the initial state x_0, there are two ways to obtain the future state x_k after k steps:

$$x_k = \Psi^{-1}(\underbrace{\cdots \hat{\mathcal{K}}_d \Psi(\Psi^{-1}(\hat{\mathcal{K}}_d}_{k \text{ times}} \Psi(x_0, u_0, \rho_0)), u_1, \rho_1) \cdots) \tag{12}$$

as shown in Fig. 1c.

For the purposes of this paper, we focus on the dynamic system with control input and the approximate Koopman operator for practical implementation. Therefore, we recommend readers who are interested in mathematical details or other types of Koopman operators to refer to [15], [16].

B. Synthesis of Koopman Basis Function

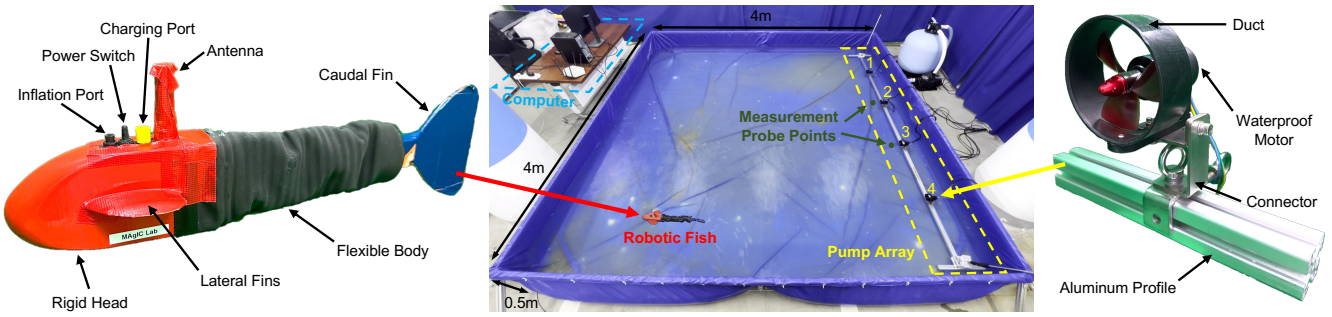
As we mention above, the representational ability of approximate Koopman operator $\hat{\mathcal{K}}_d$ stems from the fact that the basis function $\Psi(\cdot)$ provides nonlinear combinations of the original state x, which resulting in an approximate linear representation in the latent space after lifting the dimension of states x. Therefore, selecting an appropriate basis function $\Psi(\cdot)$ is the crucial key to achieve a satisfactory linear representation.

Therefore, we use the combination of polynomial and Fourier basis function to form the basis function. Specifically, the basis function in this paper is as

$$\begin{aligned}\psi_{1\sim 65}(x, u, \rho) &= \Xi(SET\{x, u, \rho\}) \\ \psi_{66\sim 77}(x, u, \rho) &= \sin(SET\{x, u, \rho\}) \\ \psi_{78\sim 89}(x, u, \rho) &= \cos(SET\{x, u, \rho\})\end{aligned} \tag{13}$$

where $SET\{\cdot\}$ means a set of all elements of vector x, u and ρ and Ξ represent a combination multiplication that selects two elements from a set to multiply. In this way, there are 89 additional basis functions, such that $\Psi(\cdot) \in \mathbb{R}^{100}$.

[1]The restore operation $\Psi^{-1}(\cdot)$ is not the mathematical inverse of $\Psi(\cdot)$ indeed, we use this symbol for clarity.



(a) (b) (c)

Fig. 2. **The Experimental Platform.** (a) The robotic fish, developed by the MAgIC Lab at ShanghaiTech University, consist of rigid head, flexible body and caudal fin. (b) The pool with flow generator, with a overhead camera at the top of the pool used to capture the motion of the robotic fish. (c) The flow generate unit, consisted of the waterproof motor with mini duct.

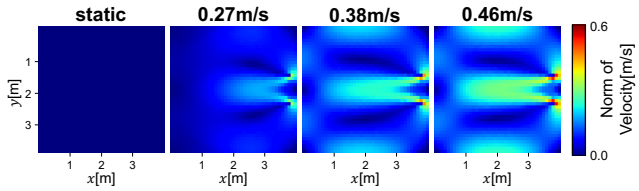


Fig. 3. **The Different Background Flow Environments.** The norm of velocity field in different background flows.

IV. Experimental Results

To evaluate the effectiveness of the proposed approach, we conducted several experiments on the developed physical platform. Specifically, we firstly deploy the robotic fish to collect motion data under several different background flow velocities. Then, we obtain the FARM model $\hat{\mathcal{K}}_d$ using the collected motion data set. Finally, the prediction accuracy of the obtained model was evaluated using unseen random motion trajectories under unseen background flow velocities.

A. Experimental Platform

As shown in Fig. 2a, the developed untethered robotic fish with length and weight are $0.51\,\mathrm{m}$ and $1.046\,\mathrm{kg}$, which consist of an ESP32S3 Micro Control Unit (MCU), a $800\,\mathrm{mAh}$ $7.4\,\mathrm{V}$ aircraft lithium battery, and one activated $5\,\mathrm{V}$ servo motor controlled according to Eq. (2). To generate different background flow fields, we built an background flow pool consists of a pool and a waterproof pump array as show in Fig. 2b and Fig. 2c. The swimming pool is in size of $4\,\mathrm{m} \times 4\,\mathrm{m} \times 0.5\,\mathrm{m}$. During the experiments, the water depth was $0.3\,\mathrm{m}$. The RGB camera is mounted upon the pool to collect motion data of the robotic fish. The computer takes charge of detecting the position and direction of the robotic fish from the captured images, while the control instructions calculated by the host computer are sent to the robotic fish wirelessly.

B. Motion Data Collection

In this work, we collect random motion data of robotic fish under four different background flow fields. The background flow of these environments are generated by driving both second and third motors with different Pulse Width Modulation (PWM) signal. We use the flow velocity values measured $5\,\mathrm{cm}$ in front of the motors to refer to corresponding fields, i.e., $0\,\mathrm{m/s}$, $0.27\,\mathrm{m/s}$, $0.38\,\mathrm{m/s}$, and $0.46\,\mathrm{m/s}$. In order to collect continuous trajectory data with sufficient length instead of being directly blown to the edge of the pool due to the background flow, we designed a proportional controller for robotic fish to track the target point P that will randomly change position every $10\,\mathrm{s}$. The control law is as

$$\alpha_a = k_a d \quad \alpha_b = k_b \beta \tag{14}$$

where $k_a = 0.5$ and $k_b = 0.2$ are control gain, and the d and β are distance and heading angle between the robotic fish and target point P, respectively. The α_f was fixed at 2π rad/s. Due to the physical limitations of robotic fish, α_b was limited to $[-65°, 65°]$ and α_a was limited to $[0°, 15°]$. During the data collection, the camera system record the position and orientation of robotic fish at $5\,\mathrm{Hz}$, and the updating frequency of the control law is $1\,\mathrm{Hz}$, which is enough for the single joint tail swinging robotic fish.

In addition, we also need the average velocity of the local flow field. As we mentioned earlier in Section II, there are currently many methods [19] that can obtain local average velocity, which is not the core of this work. Therefore, for convenience, we use COMSOL to simulate the global velocity field of the real environments, as shown in Fig. 3. After obtaining global velocity field, the average velocity of local flow field can be calculated, which is the average value of velocity of all sampling points whose distance from the robotic fish is less than $R = 0.5\,\mathrm{m}$, which is empirically taken based on the size of the robotic fish. Note that there is a difference between the velocity field calculated by simulation software and the actual velocity field indeed, but subsequent experimental results had shown that even rough velocity field information can improve the accuracy of dynamic prediction to a certain extent.

C. Training and Evaluation

We use a total of 2000 random motion data collected at $0\,\mathrm{m/s}$, $0.27\,\mathrm{m/s}$ and $0.38\,\mathrm{m/s}$ as the training data sets. According to the proposed training process as shown in Fig. 1, the heat map of the trained FARM model is shown in the Fig. 5. We can see that the FARM model is sparse, and many values in the matrix are close to zero, which means

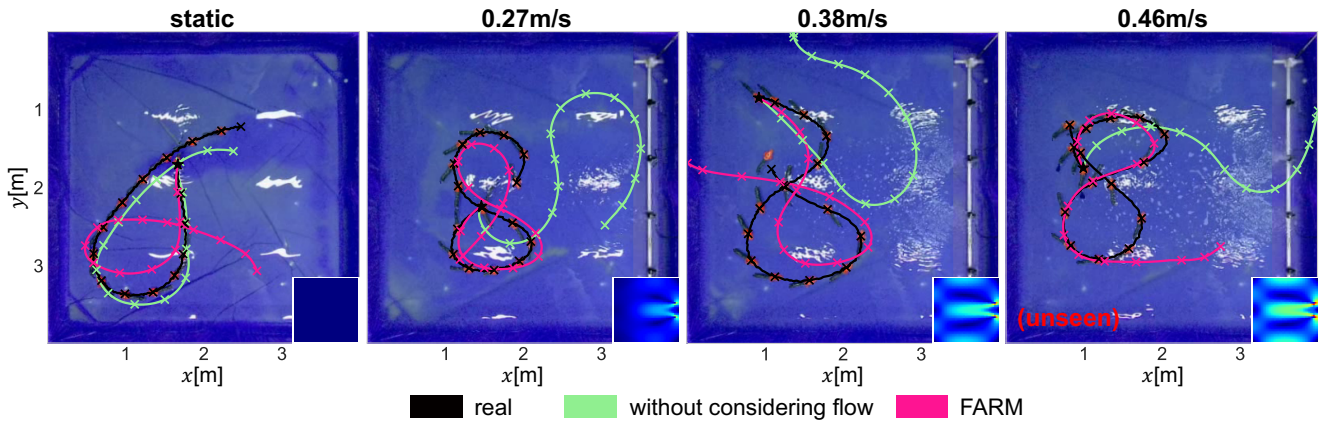


Fig. 4. **The Trajectory Result.** The long horizon prediction in different background flow condition. Each trajectory lasts for 60 s i.e., 300 timesteps of data. The position of the robotic fish is marked by a cross pattern every 4 s, while the pentagram pattern represent the initial position of each trajectory.

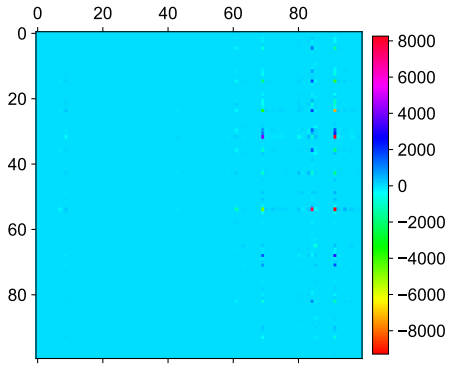


Fig. 5. **The Obtained Matrix of the FARM Model.** The heatmap of the approximate operator matrix $\hat{\mathcal{K}}_d$. Each row represents the linear combination between each observables at the next step and the current step.

the lifted functions still has room for simplification, which can lead to a lower dimensional FARM model. In addition, we also trained a baseline that only used data collected in $0\,\mathrm{m/s}$ environment, which is used to represent the dynamic model which does not consider background flow.

We collected extra new $60\,\mathrm{s}$ data at $0\,\mathrm{m/s}$, $0.27\,\mathrm{m/s}$, $0.38\,\mathrm{m/s}$ and $0.46\,\mathrm{m/s}$, and use baseline and FARM to generate predicted trajectories, as shown in Fig. 4. Note that the test motion data was not used in training phase and the data from $0.46\,\mathrm{m/s}$ even never seen before. The results shown that both the baseline and FARM had good prediction results in environment without background flow. In $0.27\,\mathrm{m/s}$ and $0.38\,\mathrm{m/s}$ environments, the predicted trajectory of FARM was closer to the ground truth trajectories compared to the baseline. Moreover, the prediction result of baseline in *unseen* $0.46\,\mathrm{m/s}$ environment is quickly diverged, while FARM still kept acceptable prediction. The above results clearly indicate that the FARM model has effectively integrated background flow velocity into the dynamic system of robotic fish. It is worth noting that the prediction results of FARM in a stationary flow field environment are slightly worse than the baseline. We believe that the main reason is that the global flow field information used for training is roughly estimated through COMSOL rather than actual measurements.

To clearly demonstrate the predictive accuracy of each element in the state, we additionally collected 9 trajectories at $0\,\mathrm{m/s}$, $0.27\,\mathrm{m/s}$, $0.38\,\mathrm{m/s}$ and $0.46\,\mathrm{m/s}$, with each one lasting for $20\,\mathrm{s}$. The absolute error between each predicted state and the actual state was recorded as shown in Fig. 6, and the Table I shows the statistical comparison of the prediction results in terms of Root Mean Square Error (RMSE).

TABLE I

THE COMPARISON RESULTS IN RMSE

Env	Method	x	y	θ	v_x	v_y	ω
0	baseline	**0.13**	**0.12**	**0.19**	**0.037**	**0.037**	**0.39**
	FARM	0.25	0.19	0.23	0.042	0.038	0.40
0.27	baseline	0.73	0.34	0.74	0.093	0.062	0.42
	FARM	**0.21**	**0.19**	**0.44**	**0.057**	**0.047**	**0.41**
0.38	baseline	0.75	0.28	0.8	0.105	0.062	0.41
	FARM	**0.28**	**0.19**	**0.37**	**0.051**	**0.048**	**0.40**
0.46	baseline	0.89	0.47	0.92	0.117	0.073	0.41
	FARM	**0.56**	**0.34**	**0.52**	**0.083**	**0.053**	**0.40**

As can be seen that the baseline performed well in environment without background flow, but its performance sharply decreased against flow conditions. The FARM had the same performance as baseline in environment without background flow and better prediction performance in environments with background flows. FARM exhibited almost identical performance in various background flows, with only slight decrease in performance in $0.46\,\mathrm{m/s}$ flow condition. In addition, we can clearly see that the prediction error of the baseline on v_x is larger than v_y, because the generated background flow is superimposed in the x-direction.

V. CONCLUSIONS

In this paper, we incorporated background flow information into the dynamic model of robotic fish utilizing the approximate Koopman operator approach. The obtained model is linear finite dimension and establishes the relationship between the average flow velocity of the local background flow field and the dynamics of the robotic fish, which greatly improved the accuracy of the dynamic model of robotic fish in

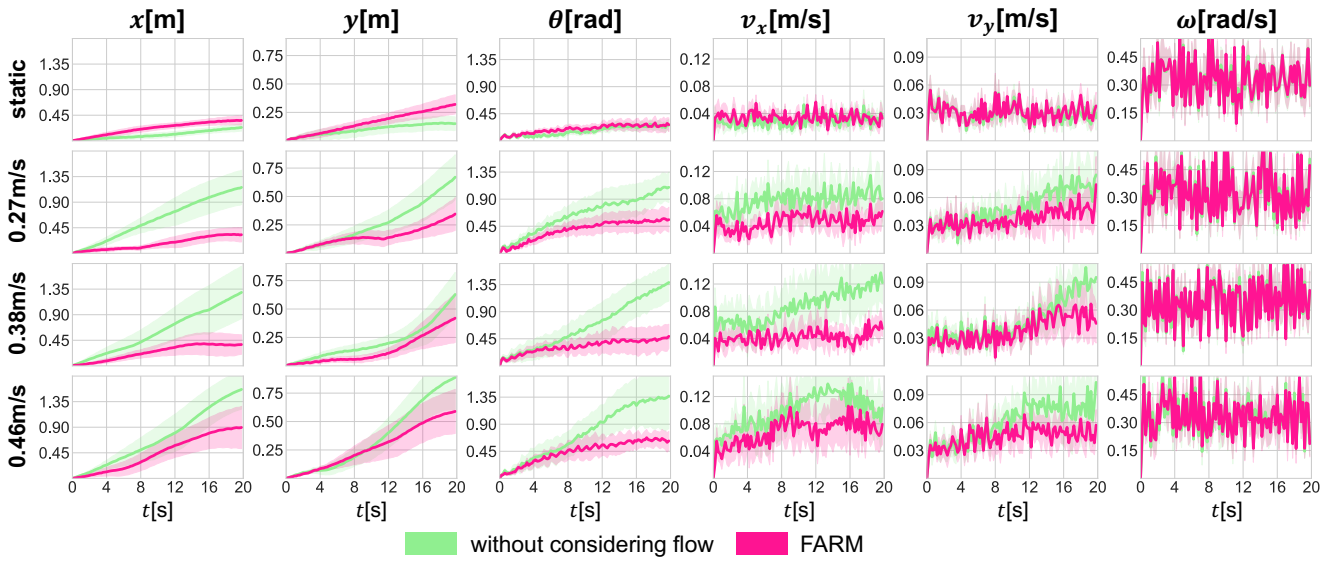


Fig. 6. **The Prediction Error Curves.** The prediction error curves of each states of robotic fish in different background flow fields. Different rows represent different background flow conditions, and different columns represent different states. In each sub-figure, the x-axis represent timestep, and the y-axis represent corresponding error values. Each case is conducted with 9 trajectory consisted of 20 s, while the solid lines representing the mean of multiple trajectories and shaded areas representing the standard deviation.

the environment with background flow without compromising the convenience of controller design. The experiments were conducted by developed robotic fish in different background flow environments and the results showed that the short- and long-term prediction performance of proposed model was satisfactory, even in unseen environment.

For future works, we will further validate the proposed dynamic modeling framework in more sophisticated environment by combination with advanced model-based controllers to achieve the challenging robotic fish control task in complex background flow.

References

[1] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," *Science Robotics*, vol. 3, no. 16, p. eaar3449, 2018.

[2] Q. Zhong, J. Zhu, F. E. Fish, S. J. Kerr, A. Downs, H. Bart-Smith, and D. Quinn, "Tunable stiffness enables fast and efficient swimming in fish-like robots," *Science Robotics*, vol. 6, no. 57, p. eabe4088, 2021.

[3] T. Zhang, R. Tian, H. Yang, C. Wang, J. Sun, S. Zhang, and G. Xie, "From simulation to reality: A learning framework for fish-like robots to perform control tasks," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3861–3878, 2022.

[4] W. Wang, X. Dai, L. Li, B. H. Gheneti, Y. Ding, J. Yu, and G. Xie, "Three-dimensional modeling of a fin-actuated robotic fish with multimodal swimming," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1641–1652, 2018.

[5] T. J. Ng, N. Chen, and F. Zhang, "Snapp: An agile robotic fish with 3-d maneuverability for open water swim," *IEEE Robotics and Automation Letters*, 2023.

[6] X. Zheng, M. Wang, J. Zheng, R. Tian, M. Xiong, and G. Xie, "Artificial lateral line based longitudinal separation sensing for two swimming robotic fish with leader-follower formation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2539–2544.

[7] X. Zheng, W. Wang, M. Xiong, and G. Xie, "Online state estimation of a fin-actuated underwater robot using artificial lateral line system," *IEEE Transactions on robotics*, vol. 36, no. 2, pp. 472–487, 2020.

[8] T. Zhang, Y. Li, S. Li, Q. Ye, C. Wang, and G. Xie, "Decentralized circle formation control for fish-like robots in the real-world via reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8814–8820.

[9] S. Yan, Z. Wu, J. Wang, M. Tan, and J. Yu, "Efficient cooperative structured control for a multijoint biomimetic robotic fish," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2506–2516, 2020.

[10] J. Yu, J. Yuan, Z. Wu, and M. Tan, "Data-driven dynamic modeling for a swimming robotic fish," *IEEE Transactions on industrial electronics*, vol. 63, no. 9, pp. 5632–5640, 2016.

[11] J. Wang and X. Tan, "Averaging tail-actuated robotic fish dynamics through force and moment scaling," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 906–917, 2015.

[12] G. Chen, X. Yang, Y. Xu, Y. Lu, and H. Hu, "Neural network-based motion modeling and control of water-actuated soft robotic fish," *Smart Materials and Structures*, vol. 32, no. 1, p. 015004, 2022.

[13] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Local koopman operators for data-driven control of robotic systems," in *Robotics: science and systems*, 2019.

[14] M. L. Castaño, A. Hess, G. Mamakoukas, T. Gao, T. Murphey, and X. Tan, "Control-oriented modeling of soft robotic swimmer with koopman operators," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 1679–1685.

[15] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[16] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.

[17] J. Zheng, T. Zhang, C. Wang, M. Xiong, and G. Xie, "Learning for attitude holding of a robotic fish: An end-to-end approach with sim-to-real transfer," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1287–1303, 2021.

[18] X. Lin, W. Song, X. Liu, X. He, and Y. Wang, "Exploring learning-based control policy for fish-like robots in altered background flows," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2338–2345.

[19] W. Zuo, F. Zhang, and Z. Chen, "Motion tomography performed by robotic fish with active heading control," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 551–556.

[20] Z. Zhang, C. Zhou, L. Cheng, X. Wang, and M. Tan, "Real-time velocity vector resolving of artificial lateral line array with fishlike motion noise suppression," *IEEE Transactions on Robotics*, 2023.

[21] D. A. Haggerty, M. J. Banks, E. Kamenar, A. B. Cao, P. C. Curtis, I. Mezić, and E. W. Hawkes, "Control of soft robots with inertial dynamics," *Science robotics*, vol. 8, no. 81, p. eadd6864, 2023.