

Dynamic Modelling of Robotic Fish with Consideration to Background Flow Using Koopman Theory

Xiaozhu Lin, Song Liu, *Member, IEEE*, 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 modelling, 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 with well consideration to background flow, i.e., Flow-Aware Robot-Fish Model (FARM), aiming at solving the non-linear and high-dimensional challenge induced by the interaction of a robotic fish with complicated environments. Specifically, we first create a Computational Fluid Dynamics (CFD)-based simulation environment for the laboratory tank that enables the generation of various flow conditions; then, by integrating flow field data provided by the simulator with real-world swimming data, we effectively employ the Koopman operator to develop a linear dynamic model for the robotic fish to represent the dynamic behaviour of a robotic fish swimming in diverse background flow scenarios. The model's efficacy with respect to conventional modelling approach in terms of behaviour prediction is validated through extensive experiments.

I. INTRODUCTION

Robotic fish has gained substantial attention in robotics field [1]–[3] in past decades due to their unique benefits, including concealment, flexibility, and energy efficiency. Notable advancements have been achieved in various aspects of this domain, including the realm of electromechanical construction [4], [5], underwater perception [6]–[8] and control [4], [9]–[12]. However, the problem of dynamic modelling [13]–[15] for the robotic fish, although plays a crucial role in motion planning and control, remains unsolved and challenging due to its inherently high-dimensional and non-linear nature, particularly when operating in complicated environments with background flow.

When it comes to dynamic modelling like for robotic fish, the fundamental approach is to derive them from first principles [13], [16], [17]. However, such solutions usually undergo significant simplification [13], [17], hence demanding complex expert knowledge and may lead to an inaccurate model. Furthermore, to obtain an accurate model, a large amount of real-world data to identify the unknown parameter [16] is still mandatory. Inspired by this fact, several works [14], [18], [19] have circumvented any a priori 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 work was in part supported by the National Natural Science Foundation of China under Grant 62303321.

X. Lin, S. Liu, and Y. Wang are with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. {linxzh, liusong, wangyang4}@shanghaitech.edu.cn

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.* [15], [20] and Castano *et al.* [21] used the Koopman operator [22]–[24] 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 problem, they share a common limitation: background flow is not well considered during dynamic modelling. Due to the absence of flow field consideration in the modelling process, the established dynamic models are likely to exhibit inaccuracies when applied complicated environments with background flow, such as in ocean and river. Given the complex nature of background flow, researchers are often compelled to resort to end-to-end learning of intricate flow fields [25]. These solutions circumvent the modelling problem of robotic fish in complicated environments, but the reliability and generality of the obtained control policy are limited.

In summary, the dynamic modelling problem for robotic fish with background flow remains challenging. A notable fact also expedites the urgency to solve this challenge, that is numerous bio-mimetic sensors [25], [26], such as the Artificial Lateral Line System (ALLS) system, have begun to emerge for sensing the flow field surrounding robotic fish. To this end, a novel dynamic modelling framework, named Flow-Aware Robot-fish Model (FARM), is proposed in this work. First, we create a Computational Fluid Dynamics (CFD)-based virtual environment for the laboratory tank that enables the generation of non-stationary flow conditions. Subsequently, by integrating flow field data provided by the simulator with real-world swimming data, we effectively employ the Koopman operator to develop a linear dynamic model for the robotic fish. This model is well-suited for representing the dynamic behavior of a robotic fish swimming in diverse background flow scenarios. The model's efficacy is validated through an extensive performance prediction experiment and comparison to conventional method without consideration to background flow.

The rest of the paper is organized as follows. We first give the problem statement in Section II. In Section III, we elaborate the approach of dynamic modelling of robotic fish. The evaluation of proposed model is presented in Section IV. Finally, we conclude the experimental results and look ahead with future work in Section V.

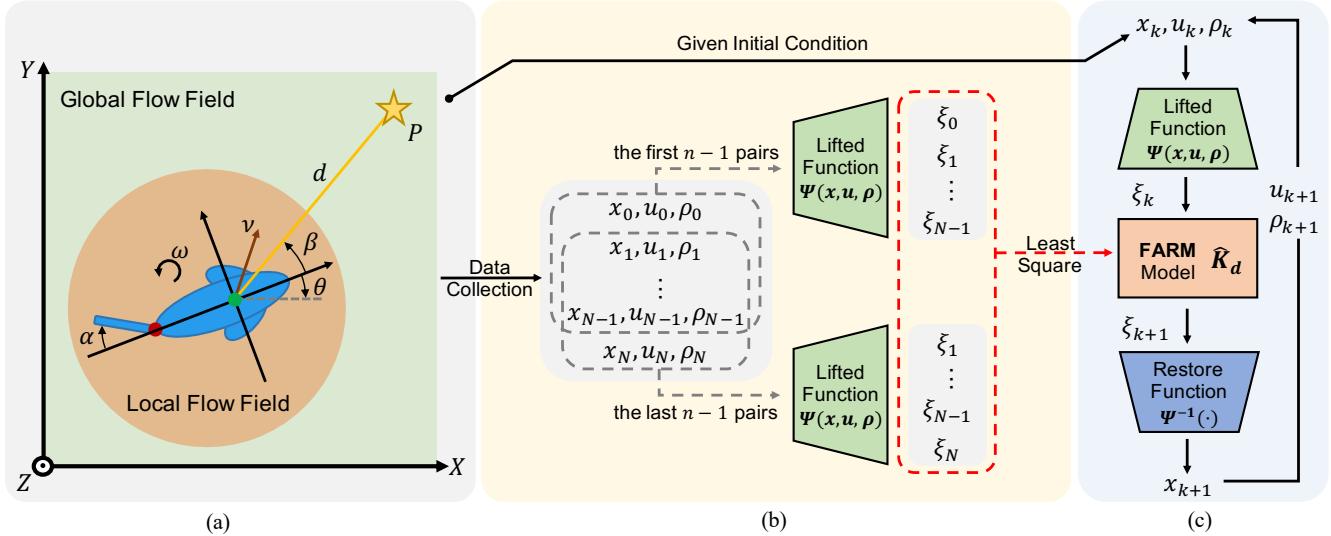


Fig. 1: Overview of Pipeline. (a) Schematic diagram of robotic fish undergoing planar locomotion with background flow. (b) Training the Koopman operator using customized lifted function with collected real world motion data. (c) Prediction of next state using trained Koopman model given current state,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

II. FLUID-AWARE ROBOT-FISH MODEL (FARM)

Establishing the dynamic model of robotic fish is challenging, due to the high dimensional nonlinear hydrodynamics and complex geometric shape. The majority of existing works [13], [16], [20], [21] involves making basic assumptions about hydrodynamics and geometric shapes to obtain the model with undetermined parameters, and then obtaining an accurate dynamic model $F(\cdot)$ through the configuration parameters of robotic fish and/or real-world motion data of robotic fish in a tank. Usually, the dynamic model $F(\cdot)$ is given in the form of

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

where x stands for the state of robotic fish e.g. position, velocity and attitude, while u is usually the control signal of joints of robotic fish. This type of model tends to describe the relationship between the control effort of the joints and the motion of the robotic fish.

However, the dynamic model Eq. (1) possesses a critical limitation in that it is only applicable in scenarios where the robotic fish swim in an environment with negligible background flow. It is very intuitive that once the environment has a background flow, the model will no longer be applicable. In this paper, we tend to take into account a swimming environment with background flow, and then integrate the local flow field information surrounding the robotic fish into the dynamic model of the robotic fish. To propose a more general model, we assume that the local flow field surrounding the robotic fish is uniform and denoted by ρ_k . Thus, the FARM model $F(\cdot)$ is proposed as

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

Compared to the conventional dynamic model, the FARM model proposed in Eq. (2) may have higher dimensions and more complex nonlinear relationships. Therefore, 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. 1. Specifically, 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. As with most works, the tail deflection angles of robotic fish can be controlled by periodic signals [13], [21] or Central Pattern Generator (CPG) [14], [16]. Here, the angle α is controlled by the periodic function as

$$\alpha(t) = \alpha_b + \alpha_a \sin(\alpha_f * t) \quad (3)$$

where α_a , α_b , α_f are denoted as amplitude, bias, and frequency of the periodic function, respectively. Based on above statement, the specific content of each variable in Eq. (2) follows as

$$\begin{aligned} x_k &= [p_x, p_y, \theta, v_x, v_y, \omega]^T \\ u_k &= [\alpha_a, \alpha_b, \alpha_f]^T \end{aligned} \quad (4)$$

Finally, the local flow field surrounding the robotic fish can be described by the velocity $\rho = [\rho_x, \rho_y]$.

It is worth noting that even without considering the background flow, the robotic fish dynamic model obtained in the aforementioned literature [14], [16] is challenging to analyze and design controllers for due to its inherent nonlinearity and black-box nature. For this reason, we use the Koopman operator [24] in this paper, to obtain a linear approximation of the dynamic model in Eq. (2). Unlike the linearization in equilibrium points, the linear representation obtained by the Koopman operator is a global linear approximation. The specific description of modelling approach will be elaborated in the Section III.

III. DYNAMIC MODELLING WITH KOOPMAN OPERATOR

In this paper, we use the Koopman operator to obtain the FARM model as shown in Fig. 1 (b). Koopman is a linear infinite dimensional operator which can transform a nonlinear finite dimensional system into a linear but infinite dimensional system. In practical use, finite dimensional approximation is used to obtain a tractable system.

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 $\rho_k \in \mathbb{R}^2$. The Koopman operator is an infinite-dimensional operator that can express a finite-dimensional nonlinear system as a linear one, which is defined as

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

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

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

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.

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

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

$$\begin{aligned}X &= [x_1, \dots, x_P] \\ U &= [u_1, \dots, u_P] \\ \Omega &= [\rho_1, \dots, \rho_P]\end{aligned}\quad (8)$$

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

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

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

$$\hat{\mathcal{K}}_d = \mathcal{A}\mathcal{B}^\dagger \quad (10)$$

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

where \dagger denotes the Moore-Penrose pseudo-inverse, and

$$\begin{aligned}\mathcal{A} &= \frac{1}{P} \sum_{k=1}^{P-1} \xi_{k+1} \xi_k^T \\ \mathcal{B} &= \frac{1}{P} \sum_{k=1}^{P-1} \xi_k \xi_k^T\end{aligned}\quad (11)$$

As we mention above, the representational ability of Koopman operator stems from the fact that the lifted 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. Therefore, selecting an appropriate lifted 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 lifted function according to [27]. Specifically, the lifted function in this paper is as

$$\begin{aligned}\psi_{1 \sim 65}(x, u, \rho) &= \Xi(SET\{x, u, \rho\}) \\ \psi_{66 \sim 77}(x, u, \rho) &= \sin(SET\{x, u, \rho\}) \\ \psi_{78 \sim 89}(x, u, \rho) &= \cos(SET\{x, u, \rho\})\end{aligned}\quad (12)$$

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 lifted functions, such that $\Psi(\cdot) \in \mathbb{R}^{100}$.

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:

- **Restored:** Restore back to the original state space before next propagation in the lifted state space.

$$x_k = \Psi^{-1}(\underbrace{\cdots \hat{\mathcal{K}}_d \Psi(\Psi^{-1}(\hat{\mathcal{K}}_d \Psi(x_0, u_0, \rho_0)), u_1, \rho_1) \cdots}_{k \text{ times}}) \quad (13)$$

- **Unrestored:** Propagate k times directly in the lifted state before restore back to the original state space.

$$x_k = \Psi^{-1}(\underbrace{\hat{\mathcal{K}}_d^k \Psi(x_0, u_0, \rho_0)}_{k \text{ times}}) \quad (14)$$

and we propagate the dynamics according to Eq. (13) in this paper as shown in Fig. 1 (c).

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 [24], [27], [28].

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness of FARM, we considered environments with four background flow conditions. One of them is with static background flow, while the other three with different background flows. Specifically, we first deployed the developed robotic fish to randomly swim in the pool to collect the trajectory of the robotic fish under different background flows; afterwards, we calculated the FARM model $\hat{\mathcal{K}}_d$ using the collected data; finally, we evaluated

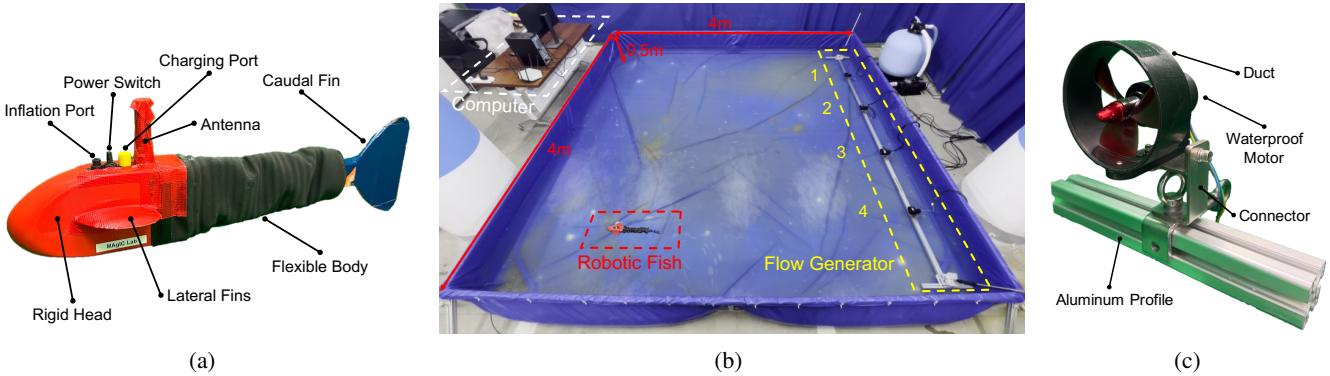


Fig. 2: **The Experimental Platform.** (a) The robotic fish, developed by the MAgIC Lab at ShanghaiTech University, consist of rigid head, flexible body and caudal fin. (b) The pool with flow generator, with an overhead camera at the top of the pool used to capture the motion of the robotic fish. (c)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

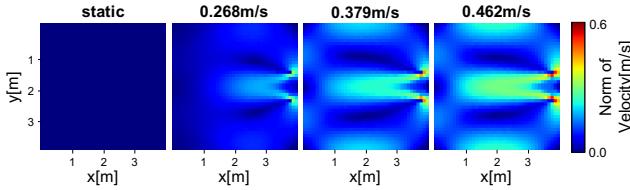


Fig. 3: **The Different Background Flow Environments.** The norm of velocity field in different background flows.

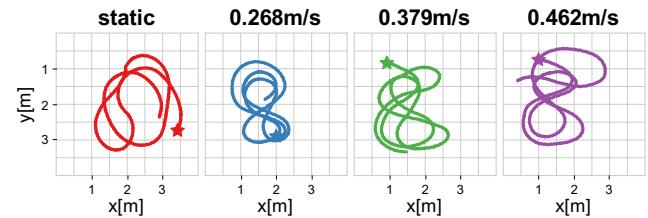


Fig. 4: **The Random Trajectories.** One of random trajectories collected in different background flows.

the prediction performance of the obtained dynamic model in different background flow conditions. We also compared FARM with a baseline [16], [20] model, trained using the data without background flow, to highlight the superiority of FARM.

A. Experimental Setup

1) *The Robotic Fish:* We conducted experiments with the developed robotic fish as shown in Fig. 2 (a), whose length and weight fish is 0.51 m and 1.046 kg. The untethered robotic fish has two fixed lateral fins for dynamic balance and an ESP32S3 Micro Control Unit, powered by a 800 mAh 7.4 V aircraft lithium battery inside the head. The robotic fish has three articulated joints in the tail, actuated by a 5 V servo motor.

2) *The Swimming Pool:* To obtain different flow conditions, we built an equipment that can generate different flows, consisting of a swimming pool and a waterproof pump array as shown in Fig. 2 (b) and Fig. 2 (c). The swimming pool is in size of 4 m × 4 m × 0.5 m. During the experiments, the water depth was 0.3 m.

3) *The Camera System:* In order to collect motion data of robotic fish, a RGB recording camera is mounted upon the swimming pool. An host computer takes charge of detecting the position and orientation of the robotic fish from the captured images. Control instructions calculated by the host computer are sent to the robotic fish wirelessly to change the swinging mode of the robotic fish.

B. Environments with Background Flow Conditions

In this work, we consider environments with four different background flow conditions i.e., **0 m/s**, **0.268 m/s**, **0.379 m/s**, and **0.462 m/s**, corresponding to the measured flow velocity from the pumps. To evaluate the performance of the dynamic model in terms of state prediction ability, the global flow velocity of background flow in the swimming pool is necessary thereof. For this reason, we used the COMSOL Multiphysics to numerically calculate the global velocity field, which can be hardly measured in real world. The global velocity ρ can be represented as

$$\rho = \bar{\rho} = \frac{1}{C} \sum_{i=1}^C \rho_i \quad (15)$$

where ρ_i is the flow velocity of i th sampling point. The distance between these sampling points and the centroid of the robotic fish was less than 0.5 m.

It should be noted that there are certain differences between the velocity field in real environment and the one from simulation, inevitably resulting in model inaccuracy. However, getting accurate flow distribution is out of the scope of this work. The experimental results of the established dynamic model with simulated velocity fields also showed up good motion prediction performance, which in turn validated the versatility of the dynamic modelling framework in practice.

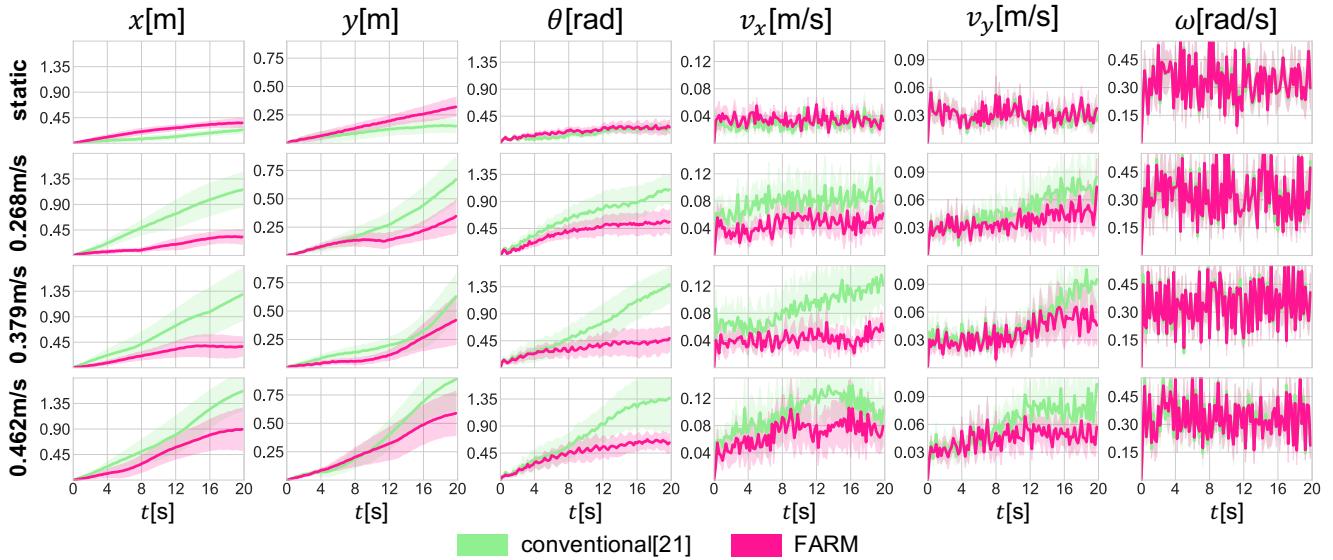


Fig. 5: The Error Curves of Prediction. The prediction error curves of various states of robotic fish in different background flows. Different rows represent different background flow conditions, and different columns represent different states of robotic fish. In each sub-figure, the x-axis represent timestep, and the y-axis represent corresponding error values. Each case is conducted with 9 trajectory consisted of 20 s data. The solid lines representing the mean of multiple trajectories and shaded areas representing the standard deviation.

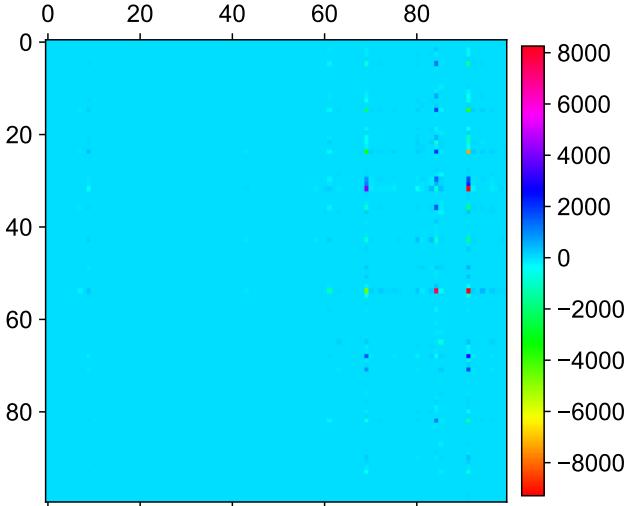


Fig. 6: The Heat Map of FARM.

C. Training Data Collection

From the literature [16], [20], [21], motion data x_k, u_k from real world were collected by applying random control input or certain fixed parameters to the robotic fish. However, this method dose not work in environment with background flow. The robotic fish will soon lose control and collide with the boundary of the swimming pool, making it impossible to collect continuous motion data with sufficient length for subsequent modelling or evaluation. For this reason, we designed a trivial proportion control law for robotic fish to tracking a random target points P as shown in Fig. 1. Random target points satisfy a two-dimensional average distribution in the swimming pool, and change position every

10 s. Specifically, during the data collection process, the control input was calculated by the following control law as

$$\begin{aligned}\alpha_a &= k_a d \\ \alpha_b &= k_b \beta\end{aligned}\quad (16)$$

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 $[-65^\circ, 65^\circ]$ and α_a was limited to $[0^\circ, 15^\circ]$.

During the data collection, the overhead camera captured the position and orientation of robotic fish at 5 frames per second (FPS), the updating frequency of the control law was 1 Hz. We collected 4 random trajectories in each environment, with each trajectory lasting for 3 minutes, to calculate dynamic model. Fig. 4 shows one of 4 random trajectories in each environment, illustrating the impact of background flow on trajectories of robotic fish.

According to the above details, training data for FARM model is collected from **0 m/s**, **0.268 m/s** and **0.379 m/s** environments, while the training data for baseline model is only collected from **0 m/s** environment.

D. Motion Prediction Evaluation of FARM

The established FARM for the experiment setup was graphically depicted by Fig. 6. The accuracy of a dynamic model is characterized by the ability of motion prediction over time. To this end, we additionally collected 9 trajectories in environments with background flow conditions of **0 m/s**, **0.268 m/s**, **0.379 m/s**, and **0.462 m/s**, with each one lasting for 20 s. Note that the motion data mentioned here was not used in calculating the dynamic model before and the data from **0.462 m/s** even never seen before. Following the

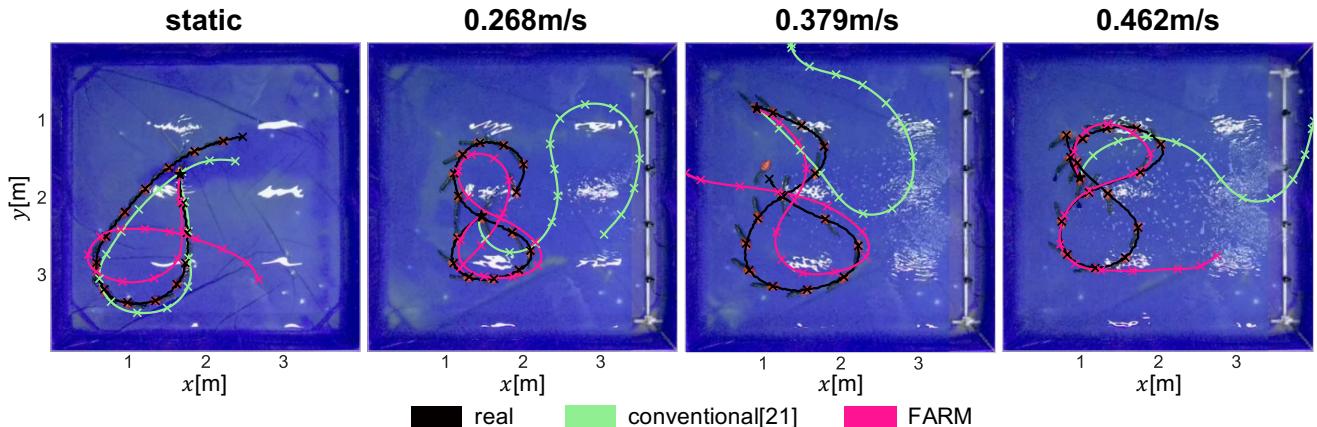


Fig. 7: **The Trajectory Result.** The long horizon prediction in different background flow condition. Each trajectory lasts for 60 s i.e. 300 timesteps of data. The position of the robotic fish is marked by a cross pattern every 4 s, while the pentagram pattern represent the initial position of each trajectory.

prediction process shown in Fig. 1 (c), the predicted states of the robotic fish was obtained from the established dynamic model given initial states and control input sequence. The absolute error between each predicted state and the actual state was recorded during the prediction process. Fig. 5 shows the prediction error for different states in different environments, where the solid line represents the mean and the shaded area represents the mean square deviation. 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 |
| 0.268 | 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 |
| 0.379 | 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 |
| 0.462 | 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. 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 **0.462 m/s** flow condition which was not seen during model calculation. In addition, by comparing the error curves of the velocity component v_x and v_y , we can clearly see that the prediction error of the baseline on v_x is much greater than v_y . A major reason for this phenomenon is that the generated background flow was superimposed to the x-direction as shown in Fig. 2 and Fig. 3.

To further intuitively demonstrate the prediction perfor-

mance of FARM, we collected 60 s trajectory in each environment, as shown in Fig. 7. It can be seen that both the baseline and FARM had good prediction results in environment without background flow. In **0.268 m/s** and **0.379 m/s** environments, the predicted trajectory of FARM was closer to the ground truth trajectories compared to the baseline. Moreover, the result in **0.462 m/s** environment also conformed that the trajectory predicted by the baseline model quickly diverged, while FARM still kept acceptable prediction, especially considering it was 60 s trajectory, which doubled the time length of the training trajectories.

V. CONCLUSIONS

In this paper, we incorporated background flow information into the dynamic modelling of robotic fish, which greatly improved the accuracy of the dynamic model of robotic fish, especially in environments with background flows. The experimental results were conducted by delicately developed robotic fish in swimming pool. Experimental results well conformed the effectiveness of the proposed dynamic modelling framework for robotic fish motion control in complicated environments, outperforming traditional models in terms of motion prediction. For future works, we will further validate the proposed dynamic modelling framework in more sophisticated environment by combination with advanced model-based controllers to achieve the challenging robotic fish control task in complex background flow. As an explanatory effort, this paper contributes a new insight to the fish-like robotics research community for effective motion control with high dimensional nonlinear dynamics.

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] B. Sun, W. Li, Z. Wang, Y. Zhu, Q. He, X. Guan, G. Dai, D. Yuan, A. Li, W. Cui, *et al.*, “Recent progress in modeling and control of bio-inspired fish robots.” *Journal of Marine Science and Engineering*, vol. 10, no. 6, p. 773, 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] X. Liao, C. Zhou, Q. Zou, J. Wang, and B. Lu, “Dynamic modeling and performance analysis for a wire-driven elastic robotic fish,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11174–11181, 2022.
- [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] X. Guo, S. Zhang, K. Zhou, J. Zheng, C. Wang, and G. Xie, “Azimuth estimation of swimming fish by artificial lateral line system,” in *2022 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE, 2023, pp. 216–221.
- [9] T. Zhang, L. Yue, C. Wang, J. Sun, S. Zhang, A. Wei, and G. Xie, “Leveraging imitation learning on pose regulation problem of a robotic fish,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [10] 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.
- [11] S. Yan, Z. Wu, J. Wang, Y. Huang, M. Tan, and J. Yu, “Real-world learning control for autonomous exploration of a biomimetic robotic shark,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 4, pp. 3966–3974, 2022.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] G. Mamakoukas, M. L. Castano, X. Tan, and T. D. Murphey, “Derivative-based koopman operators for real-time control of robotic systems,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2173–2192, 2021.
- [16] 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.
- [17] J. Wang, S. Chen, and X. Tan, “Control-oriented averaging of tail-actuated robotic fish dynamics,” in *2013 American Control Conference*. IEEE, 2013, pp. 591–596.
- [18] T. Zhang, R. Tian, C. Wang, and G. Xie, “Path-following control of fish-like robots: A deep reinforcement learning approach,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8163–8168, 2020.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] N. Takeishi, Y. Kawahara, and T. Yairi, “Learning koopman invariant subspaces for dynamic mode decomposition,” *Advances in neural information processing systems*, vol. 30, 2017.
- [23] A. S. Dogra and W. Redman, “Optimizing neural networks via koopman operator theory,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2087–2097, 2020.
- [24] 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.
- [25] 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.
- [26] F. Zhang, F. D. Lagor, D. Yeo, P. Washington, and D. A. Paley, “Distributed flow sensing for closed-loop speed control of a flexible fish robot,” *Bioinspiration & biomimetics*, vol. 10, no. 6, p. 065001, 2015.
- [27] I. Abraham, G. De La Torre, and T. Murphey, “Model-based control using koopman operators,” in *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, 2017.
- [28] M. Budišić, R. Mohr, and I. Mezić, “Applied koopmanism,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, 2012.