Control-oriented Modeling of Soft Robotic Swimmer with Koopman Operators

Maria L. Castaño[1], Andrew Hess[2], Giorgos Mamakoukas[3], Tong Gao[2], Todd Murphey[3] and Xiaobo Tan[1]

***Abstract*— Interest in soft robotics has increased in recent years due to their potential in a myriad of applications. A wide variety of soft robots has emerged, including bio-inspired robotic swimmers such as jellyfish, rays, and robotic fish. However, the highly nonlinear fluid-structure interactions pose considerable challenges in the analysis, modeling, and feedback control of these soft robotic swimmers. In particular, developing models that are of high fidelity but are also amenable to control for such robots remains an open problem. In this work, we propose a data-driven approach that exploits Koopman operators to obtain a linear representation of the soft swimmer dynamics. Specifically, two methodologies are explored for obtaining the basis functions of the the operator, one based on data-based derivatives estimated using high-gain observers, and the other based on the dynamics structure of a tail-actuated rigid-body robotic fish. The resulting approximate finite-dimensional operators are trained and evaluated using data from high-fidelity CFD simulations that incorporate fluid-structure interactions. Validation results demonstrate that, while both methods are promising in producing control-oriented models, the approach based on derivative estimates shows higher accuracy in state prediction.**

I. Introduction

The need for robots that are geometrically compliant yet robust has led soft robotics to emerge as a field of interest in recent years [1], [2]. While conventional robots consisting of rigid components typically present limited adaptability, soft robots possess a large number of degrees of freedom, which allows them to achieve versatile motion and dexterous manipulation. Soft-bodied animals, such as insects, snakes and fish, have been a significant source of inspiration for the development of a wide variety of biomimetic soft robots [3], [4]. Aquatic biomimetic robots have particularly become a promising application of soft robotics since, much like aquatic animals, their soft bodies can achieve high mobility, efficiency, and dexterity [5]. Their potential in applications, including inspection, search and rescue [6], environmental sampling [7], and serving as platforms to address biological questions [8], makes soft robotic swimmers a topic of interest.

Despite the promise of soft swimming robots, their complex dynamics, large number of degrees of freedom, and fluid-structure interactions have made it challenging to obtain models that are accurate yet amenable to model-based control design. Most of the work on soft aquatic robots has focused on prototype development based on a wide variety of actuation methods and materials [9]–[12]. While there has been some work on modeling, the models obtained are typically not control-oriented [12]–[17]. More importantly, however, successful models need to be robust to unknown parameters or changes of fluid environments, while also being conducive to real-time implementation [18], [19].

Koopman operators allows one to construct explicit linear representations of complex dynamical systems [20], [21]. While this method is data-driven, it has the benefit of producing explicit linear control-affine models. There has been some work on using the Koopman operator for constructing a linear representation of rigid robotic fish with basis functions derived from higher-order derivatives of the underlying nonlinear dynamic model [22].

In this paper, we propose a data-driven approach that utilizes Koopman operators to obtain linear, control-oriented models for soft robotic swimmers. In particular, we propose two different methods for constructing the derivatives-based basis functions for the Koopman operators. Specifically, one method utilizes higher-order derivatives of the measured states, which are estimated using high-gain observers, and the other assumes that the dynamics structure of the soft robotic swimmer can be approximated using a relevant averaged model of a rigid, tail-actuated robotic fish. The proposed Koopman schemes are trained and then validated using data obtained from high-fidelity CFD simulations. Validation results show that both methods are promising, but the one based on estimated derivatives demonstrates higher accuracy in predicting the robot's behavior.

The rest of the paper is organized as follows. We first review the Koopman operator in Section II. In Section III, we detail the CFD-based model of the soft robotic swimmer and the scheme for obtaining its measured states. Construction of the basis functions for the soft robotic swimmer is presented in Section IV. In Section V, we evaluate the accuracy of the resulting Koopman operator-based linear models and discuss the results. Finally, we provide some concluding remarks in Section VI.

II. Background on Koopman Operators

Expressing a system's dynamics in a linear fashion is often desirable since it eases both the investigation of the stability properties [23] and the control synthesis of complex systems [22]. Unlike linearization, which naturally becomes inaccurate away from the linearizing point, the Koopman

*This research was supported in part by the National Science Foundation (DGE1424871,IIS 1715714, IIS 1848945, CBET1702987) and an MSU Strategic Partnership Grant (16-SPG-Full-3236).

[1]Maria L. Castaño and Xiaobo Tan are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA. Email: castanom@msu.edu (M.L.C)

[2] Andrew Hess and Tong Gao are with the Department of Mechanical Engineering and the the Department of Computational Mathematics, Sciences, and Engineering, Michigan State University, East Lansing, MI 48824, USA.

[3] Giorgos Mamakoukas and Todd Murphey are with the Department of Mechanical Engineering, Northwestern University, Evanston, Illinois 60208, USA

operator is able to evolve a nonlinear system with full fidelity throughout the state space [24]. This makes Koopman operators an attractive approach for obtaining reliable linear representations of complex systems.

A. Koopman Operator

The Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator that evolves functions of the states $\xi \in \mathbb{R}^N$ of a dynamics system. These functions, $\Phi(\xi)$, often termed observables, thus evolve according to

$$\frac{d}{dt}\Phi(\xi) = \mathcal{K}\Phi(\xi) \quad \text{or} \quad \Phi(\xi_{k+1}) = \mathcal{K}_d\Phi(\xi_k) \tag{1}$$

in the continuous-time and discrete-time settings, respectively.

The infinite-dimensional nature of the Koopman operator prevents it from being practical in general. Nevertheless, there has been some progress towards obtaining finite-dimensional linear Koopman representations, which are only applicable for certain classes of nonlinear systems [25]–[30]. In the case where no finite-dimensional Koopman invariant subspace can be found, recent studies have implemented data-driven regression schemes to approximate the infinite-dimensional operator $\mathcal{K}$ with a finite-dimensional approximation $\tilde{\mathcal{K}}$, [21], [31], [32]. Although approximating the Koopman operator induces errors in the system propagation, it has been shown that the linear model is able to evolve the original system with acceptable accuracy [22]. Improving both the short- and the long-term accuracy of the approximate Koopman models is an active research topic, with positive results from recent efforts that try to impose conditions on the learned representation (such as dissipativity [33], or stability [34]) in order to more closely match properties of the underlying dynamics.

B. Data-driven Finite-dimensional Approximation

In this work we utilize the least-squares method [21] to obtain a finite-dimensional operator. In particular, one can obtain a Koopman operator approximation $\tilde{\mathcal{K}}$ by choosing a set of basis functions $\Phi(\xi)$ and solving a least-square minimization problem. To include the actuation terms, let the observables be defined as $\Phi(\xi, u) = [\phi_1(\xi, u), \phi_2(\xi, u), \phi_3(\xi, u), \ldots, \phi_q(\xi, u)]^T \in \mathbb{R}^q$. In the discrete-time case, the minimization problem takes the following form

$$\tilde{\mathcal{K}}_d = \arg\min_{\tilde{\mathcal{K}}_d} \sum_{k=1}^{P-1} \frac{1}{2} \|\Phi(\xi_{k+1}, u_{k+1}) - \tilde{\mathcal{K}}_d\Phi(\xi_k, u_k)\|^2 \tag{2}$$

where P denotes the number of measurements, which are composed of an initial state ξ_k, a final state ξ_{k+1} and the actuation applied at each instant, u_k and u_{k+1}. It was shown that in [21] that there is a closed-form solution to the problem presented above, and is given by

$$\tilde{\mathcal{K}}_d^* = \mathcal{A}\mathcal{B}^\dagger \tag{3}$$

where

$$\mathcal{A} = \frac{1}{P}\sum_{k=1}^{P} \frac{1}{}\Phi(\xi_{k+1}, u_{k+1})\Phi(\xi_k, u_k)^T \tag{4}$$

$$\mathcal{B} = \frac{1}{P}\sum_{k=1}^{P} \frac{1}{}\Phi(\xi_k, u_k)\Phi(\xi_k, u_k)^T \tag{5}$$

and $\dagger$ denotes the Moore-Penrose pseudoinverse. The continuous-time operator can be obtained via $\mathcal{K} = \log(\mathcal{K}_d)/\Delta t$, where Δt is the time spacing between measurements, ξ_k and ξ_{k+1} [21].

In the sections that follow we propose several methods for constructing the observables in order to approximate a Koopman operator for a soft robotic swimmer.

III. Soft Robotic Swimmer: CFD Modeling and Data Extraction

Soft robotic swimmers exhibit complex behavior and posses dynamics that are determined by highly nonlinear fluid-structure interactions. The authors in [35] demonstrated that the active strain approach in a distributed Lagrange multiplier/fictitious domain (DLM/FD) method can capture this complex behavior with high fidelity. In this paper, we use the computational fluid dynamics (CFD) model to generate the data used for the training and validation of the Koopman-based models.

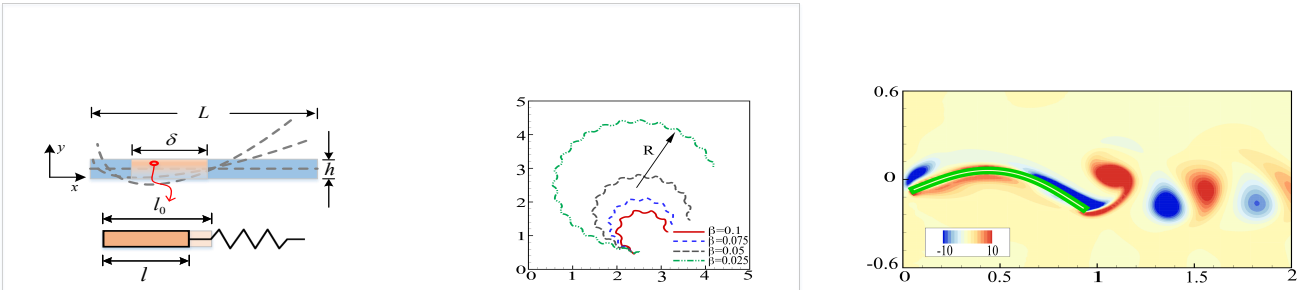
A. CFD Model

The authors in [35] recently developed a computational framework for simulation of a soft robotic swimmer that uses a fictitious domain/active-strain method to simulate soft robots' swimming motions at finite Reynolds (Re) numbers (typically in the range of 100 to 1000). The method assumes that the robot is composed of soft active materials and can perform finite/large deformations when subjected to stimulation. The soft swimmer is modeled as a 2D rectangular beam of length L and uniform thickness h made up of a continuous hyperelastic material which, at the microscopic level, is driven by a contracting element with an initial length l_0. Fig. 1(a) illustrates how the slender fish body shape is characterized. To activate the beam in a continuous and periodic fashion, a constant ($\lambda_a = 1-\alpha_0$) or time-dependent contractile strain contractile field is applied alternatively on both sides of the beam with some period T.

To control the resultant undulatory swimming motion, the magnitude of the contraction strength α is formulated as

$$\alpha_R(t) = \alpha_0(t) + \frac{\beta(t)}{2}, \quad \alpha_L(t) = \alpha_0(t) - \frac{\beta(t)}{2}, \tag{6}$$

where $\alpha_R(t)$ and $\alpha_L(t)$ correspond to the actuation strength on the right and left sides, respectively. Here $\alpha_0(t)$ refers to the "base" contraction strength to achieve forward ("straight") swimming motion and it is to be limited to some range $0 \leq \alpha_0 \leq \alpha_{max}$, and β denotes the bias between the contraction strengths of the two actuated sides which allows for turning. The soft swimmer will then turn in the direction of stronger actuation. Fig. 1(b), demonstrates that the soft swimmer turning motions with different turning radii can be



(a) (b) (c)

Fig. 1: FD/active-strain simulator: (a) Micro-mechanical model of a 2D beam under contractile actuation (view from above). The active segment is of length δ. (b) Trajectories of turning motions by tuning β. (c) Instantaneous snapshot of vorticity field of a free-swimming beam at $\mathrm{Re} = 500$ and $T = 2.0$. [35]

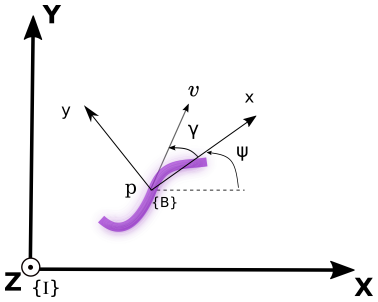


Fig. 2: Top view of the soft robotic swimmer undergoing planar motion.

effectively achieved by tuning β.

To simulate the swimming motions of soft robots, this active strain model is implemented using a fictitious domain (FD) method which solves the fully coupled nonlinear fluid/elastic-structure interactions (FESI). When coupled with the fluid-flow solver via the fictitious domain formulation, one can simulate how an active beam will undergo periodic undulatory motions as illustrated by the vorticity map in Fig. 1(c). The reader is referred to [35] for further details.

B. Representation of States

Given their large compliance, soft robots tend to have a large number of degrees of freedom (DOFs), which makes modeling very challenging. Instead of attempting to capture the evolution of each DOF, we only consider the movements of the soft body as a whole. The soft robotic swimmer is then viewed as a general 3-DOF rigid body with 2 inputs that undergoes planar motion. Let $\{\mathrm{I}\}$ denote the inertial coordinate frame and $\{\mathrm{B}\}$ the body-fixed reference frame attached to its center of mass. In particular, let $[\mathbf{X},\ \mathbf{Y},\ \mathbf{Z}]^T$ and $[\mathrm{x},\ \mathrm{y},\ \mathrm{z}]^T$ denote the inertial and the body-fixed coordinate systems, respectively, as illustrated in Fig. 2. Note that the body-fixed x axis is defined by the connecting line between the center of the swimmer and its anterior leading tip. Let x, y denote the position of the origin of $\{\mathrm{B}\}$ in $\{\mathrm{I}\}$. Furthermore, let v_1 , v_2, and ω denote the body-fixed surge, sway and angular velocities, respectively. Furthermore, let γ denote the angle of attack, formed by the direction of $\boldsymbol{v} = [v_1, v_2]^T$ with respect to the body-fixed x-axis, and let ψ denote the heading angle, formed by the body-fixed x-axis relative to the inertial $\mathbf{X}$-axis. In the next section, we will discuss how to capture the dynamics that govern the evolution of these velocity states, v_1, v_2, and ω, using Koopman operators.

IV. Synthesis of Derivative-based Koopman Basis Functions

In this section, using the system states defined previously, we propose two approaches to populate the observables of the approximate Koopman operator. Recall that in the continuous-time setting the approximate Koopman operator $\tilde{\mathcal{K}}$ evolves functions of the states and inputs of a system [22] such that

$$\frac{\mathrm{d}\Phi(\xi,u)}{\mathrm{d}t} \approx \tilde{\mathcal{K}}\Phi(\xi,u) \tag{7}$$

without loss of generality, the above can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\Phi_\xi(\xi)\\ \Phi_{\xi,u}(\xi,u)\end{bmatrix} \approx \begin{bmatrix}\tilde{\mathcal{K}}_\xi & \tilde{\mathcal{K}}_{\xi,u}\\ \tilde{\mathcal{K}}_{u,\xi} & \tilde{\mathcal{K}}_u\end{bmatrix}\begin{bmatrix}\Phi_\xi(\xi)\\ \Phi_{\xi,u}(\xi,u)\end{bmatrix} \tag{8}$$

where $\Phi(\xi,u) = [\Phi_\xi(\xi), \Phi_{\xi,u}(\xi,u)]^T$, with $\Phi_\xi(\xi) \in \mathbb{R}^{q_\xi}$ being only dependent on the states and $\Phi_{\xi,u}(\xi,u) \in \mathbb{R}^{q_u}$ being dependent of the inputs as well. Note that $q = q_\xi + q_u$. Here $\tilde{\mathcal{K}}_\xi \in \mathbb{R}^{q_\xi \times q_\xi}$, $\tilde{\mathcal{K}}_{\xi,u} \in \mathbb{R}^{q_\xi \times q_u}$, $\tilde{\mathcal{K}}_{u,\xi} \in \mathbb{R}^{q_u \times q_\xi}$ and $\tilde{\mathcal{K}}_u \in \mathbb{R}^{q_u \times q_u}$ are sub-matrices of $\tilde{\mathcal{K}}$.

By linearizing the above Koopman representation with respect to $\Phi_\xi(\xi)$ and u, one can obtain a state and control-affine linear model, such that

$$\begin{aligned}\frac{\mathrm{d}\Phi_\xi(\xi)}{\mathrm{d}t} \approx& \frac{\partial}{\partial\Phi_\xi(\xi)}\Big(\tilde{\mathcal{K}}_\xi\Phi_\xi(\xi) + \tilde{\mathcal{K}}_{\xi,u}\Phi_{\xi,u}(\xi,u)\Big)\Phi_\xi(\xi)\\ &+ \frac{\partial}{\partial u}\Big(\tilde{\mathcal{K}}_\xi\Phi_\xi(\xi) + \tilde{\mathcal{K}}_{\xi,u}\Phi_{\xi,u}(\xi,u)\Big)u\\ =&\Big(\tilde{\mathcal{K}}_\xi + \frac{\partial}{\partial\Phi_\xi(\xi)}\tilde{\mathcal{K}}_{\xi,u}\Phi_{\xi,u}(\xi,u)\Big)\Phi_\xi(\xi)\\ &+ \Big(\tilde{\mathcal{K}}_{\xi,u}\frac{\partial}{\partial u}\Phi_{\xi,u}(\xi,u)\Big)u\\ =&A(\xi,u)\Phi_\xi(\xi) + B(\xi)u\end{aligned} \tag{9}$$

where $A(\xi,u) \triangleq (\tilde{\mathcal{K}}_\xi + \tilde{\mathcal{K}}_{\xi,u}\frac{\partial}{\partial\Phi_\xi(\xi)}\Phi_{\xi,u}(\xi,u))$ and $B(\xi) \triangleq \tilde{\mathcal{K}}_{\xi,u}\frac{\partial}{\partial u}\Phi_{\xi,u}(\xi,u)$, and would be evaluated around the nominal trajectories ξ^* and u^*.

Having A and B be state-dependent requires the constant reevaluation of Eq. (9); however, by letting $\Phi_{\xi,u}(\xi,u) = u$

one can avoid this, such that

$$\begin{aligned}\frac{\mathrm{d}\Phi_\xi(\xi)}{\mathrm{d}t} \approx& \tilde{\mathcal{K}}_\xi \Phi_\xi(\xi) + \tilde{\mathcal{K}}_{\xi,u} u \\ =& A\Phi_\xi(\xi) + Bu\end{aligned} \tag{10}$$

Note that in this manner A and B are fixed and not dependent on the state and inputs. Similar analysis shows that in the discrete-time setting

$$\Phi_\xi(\xi_{k+1}) \approx A\Phi_\xi(\xi_k) + Bu(k) \tag{11}$$

By letting $\xi = [x, y, \psi, v_1, v_2, \omega]^T$, Eq. (10) can then be used to represent the soft robot dynamics with a time-invariant linear system representation by properly choosing the basis functions $\Phi_\xi(\xi)$ and the system inputs u.

From the CFD-model, there are two physical inputs at our disposal, namely the forward swimming control α_0 and the turning control β. In this work, we explore defining the system inputs u in two distinct manners: one, termed "linear inputs", treats directly α_0 and β as control inputs,

$$\begin{aligned}u_1 =& \alpha_0 \\ u_2 =& \beta\end{aligned} \tag{12}$$

and the other termed "nonlinear inputs" and inspired by the averaged robotic fish model [36], treats the system inputs as nonlinear functions of the physical variables α_0 and β, in the following form:

$$\begin{aligned}u_1 &= \alpha_0(3 - \frac{3}{2}\beta^2 - \frac{3}{8}\alpha_0^2) \\ u_2 &= \alpha_0^2 \beta\end{aligned} \tag{13}$$

Next we elaborate on the two derivatives-based methods for generating the basis functions.

A. Basis Function Synthesis Based on Estimated Higher-Order Derivatives

Without prior knowledge of the system's dynamics, the first method, termed "Higher-Order Derivatives" (HOD), the basis functions are populated using higher order derivatives of the states, which are estimated using high gain observers. Let $\xi_1 = v_1, \xi_2 = v_2, \xi_3 = \omega, \xi_4 = x, \xi_5 = y, \xi_6 = \psi$. Then the basis functions $\Phi(\xi)$ can be defined as follows

$$\Phi(\xi) = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_6 \\ \phi_7 \\ \vdots \\ \phi_{Tn+n} \\ \phi_{Tn+n+1} \\ \phi_{Tn+n+2} \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_6 \\ \frac{\mathrm{d}\xi_1}{\mathrm{d}t} \\ \vdots \\ \frac{d^T}{dt^T}\xi_n \\ u_1 \\ u_2 \end{bmatrix} \tag{14}$$

where T denotes the derivative's order and n is the number of states (6 in the current case) .

B. Basis Function Synthesis Based on Robotic Fish Dynamics

The second approach, termed "Robotic Fish-Inspired" (RFI), adopts the approach proposed in [22]. In particular, we assume the dynamics of the soft robotic swimmer have the same structure as that of an approximate averaged dynamic model of a tail-actuated rigid robotic fish [36]. Although it is known that the dynamics differ between both systems, we assume the overall model structure is sufficiently close such that the observables can then be populated with the states, and the terms involved in the first and second-order derivatives of the states. Specifically, the first 6 observables are populated with the states, while separate functions are used for the time-derivatives of each individual term that appears in the dynamics and their derivatives. For example, considering that the kinematic equation and the surge acceleration of the robotic fish are given by

$$\begin{aligned}\dot{x} =& v_1 \cos(\psi) - v_2 \sin(\psi) \\ \dot{y} =& v_1 \sin(\psi) + v_2 \cos(\psi) \\ \dot{\psi} =& \omega\end{aligned} \tag{15}$$

$$\begin{aligned}\dot{v}_1 =& \frac{m_2}{m_1} v_2 \omega - \frac{c_1}{m_1} v_1 \sqrt{v_1^2 + v_2^2} \\ &+ \frac{c_2}{m_1} v_2 \sqrt{v_1^2 + v_2^2}\, \arctan(\frac{v_2}{v_1}) + b_1 u_1\end{aligned} \tag{16}$$

where m_1, m_2, c_1, b_1 are unknown constants. The first 11 basis functions are then populated as

$$\Phi(\xi) = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \\ \vdots \end{bmatrix} = \begin{bmatrix} x \\ \vdots \\ \omega \\ v_1 \cos(\psi) - v_2 \sin(\psi) \\ v_1 \sin(\psi) + v_2 \cos(\psi) \\ v_2 \omega \\ v_1^2 \\ \vdots \end{bmatrix} \tag{17}$$

where the nonlinear terms are approximated as $\sqrt{v_1^2 + v_2^2} \approx v_1$ and $\arctan(\frac{v_2}{v_1}) \approx \frac{v_2}{v_1}$. Note that to allow the identification of unknown coefficients, each term in the derivatives is considered individually. Using the nonlinear terms that appear in the rest of the dynamic equations, one can continue to uniquely construct the basis functions until all the terms are considered (see [22]). Similarly, the rest of the basis functions are constructed using the terms that appear in the second order derivatives. For example, by considering that $\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = \frac{\mathrm{d}v_1}{\mathrm{d}t}\cos(\psi) - v_1 \sin(\psi)\omega - \frac{\mathrm{d}v_2}{\mathrm{d}t}\sin(\psi) - v_2 \cos(\psi)\omega$ and by approximating the nonlinear terms that appear in $\frac{\mathrm{d}v_1}{\mathrm{d}t}$ and $\frac{\mathrm{d}v_2}{\mathrm{d}t}$ as previously mentioned, some of the basis functions are then given by

$$\begin{bmatrix} \phi_{15} \\ \phi_{16} \\ \phi_{17} \\ \vdots \end{bmatrix} = \begin{bmatrix} v_2 \omega \cos(\psi) \\ v_1^2 \cos(\psi) \\ v_2^2 \cos(\psi) \\ \vdots \end{bmatrix} \tag{18}$$

In this way, $\Phi(\xi) \in \mathbb{R}^{60}$, which is populated with the system states as well as 52 additional scalar functions of the states.

V. Model Training and Validation Results

To illustrate and evaluate the proposed methods, we first describe the training of the Koopman operators and then compare their predictive accuracy against data sets that different from the one used to do the training.

A. Training of Koopman Operators

Each Koopman operator is trained by solving the least square problem (2) with (3), where Φ is defined using one of the two methods proposed above. In this work, we consider up to 3 orders ($P = 1, 2, 3$) of the derivatives when populating the observables using the HOD method, as we are interested in determining what effect the orders of derivatives would have on the Koopman predictive performance. To estimate the higher-order derivatives of the states, high-gain observers [37] are implemented since they offer desirable robustness and stability properties. Furthermore, since two types of system inputs (linear inputs and nonlinear inputs) are also considered, a total of 8 different approaches are explored and compared.

The training data ξ_k and ξ_{k+1} are obtained based on the data collected from the CFD simulations. In particular, the physical inputs $\alpha_0(t)$ and $\beta(t)$ are varied randomly across the allowable input space such that the training data provides rich enough exploration of the dynamical system phase space. The resulting time-averaged states are then sampled at $\Delta_t = 0.01$s. Note that $\alpha_0(t)$ and $\beta(t)$ and thus the inputs u_k and u_{k+1} are kept constant between measurements. The surge v_1 and sway v_2 velocity measurements are normalized with a velocity scale defined as the beam actuation times body length frequency, and the heading angle is calculated by considering an imaginary line connecting the midpoint of the soft robot with the the anterior leading tip of the robot.

Once the Koopman operators are trained, the state and control linearization matrices A and B are extracted. The linear system is then propagated for $T = 15$ seconds from the initial time, and the predictions are compared with corresponding data. To quantify how well the Koopman operators predict a given data set, we utilize the normalized mean square error (NMSE) defined as follows:

$$fit(i) = 1 - \frac{\|\xi_{ref}(:, i) - \xi(:, i)\|^2}{\|\xi_{ref}(:, i) - \text{mean}(\xi_{ref}(:, i))\|^2} \tag{19}$$

where $\|\cdot\|$ indicates the 2-norm of a vector, ξ_{ref} is the CFD-data measurements, and $i = 1, ...n$, where n is the number of states, and mean$(\xi_{ref}(:, i))$ represents the mean of state i. Each fit is thus a row vector of length n that represents the 'fitness "for each state, i.e it is a quantitative representation of the closeness of ξ to ξ_{ref} across the time horizon T. Note that the NMSE varies between -∞ and 1, where 1 signifies a perfect fit. Finally, we average $fit(i)$ across all states to obtain an average fit for the whole data set. Table I summarizes the percent fit for all 8 methods. Note that "HOD", "HOD-2" and "HOD-3" denote utilizing up to the 1st, 2nd, and 3rd-order derivatives for the HOD method, respectively. For the sake of illustration, Fig. 3 depicts the fitness achieved by the HOD and RFI-trained Koopman operators for the case when the nonlinear inputs are considered.

TABLE I: The level of fitness between the CFD training data and the predictions by the trained Koopman for the four different basis function synthesis methods proposed. Note that the fitness values range between -∞ and 1, with 1 being a perfect fit.

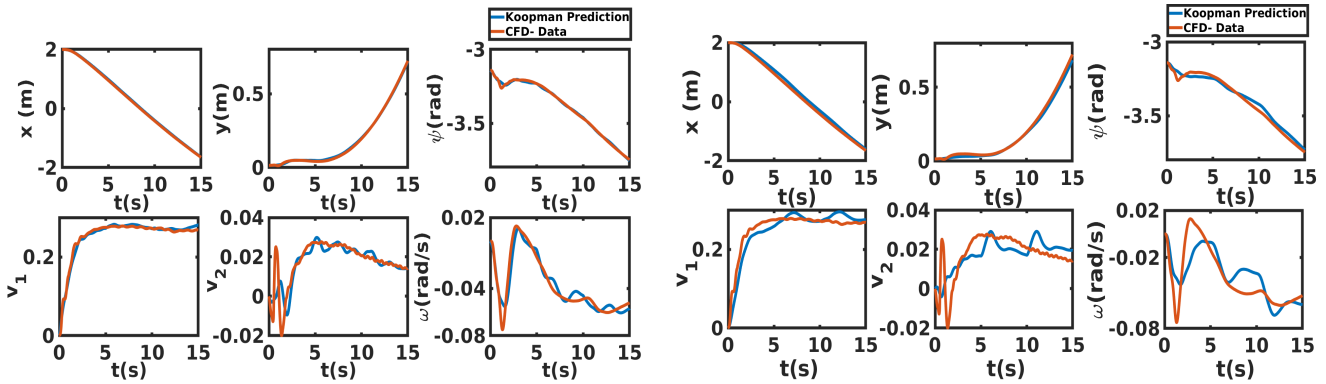
Method	Nonlinear Inputs	Linear Inputs
HOD	0.8036	0.8039
HOD-2	0.7511	0.7513
HOD-3	0.7634	0.7636
RFI	0.9315	0.9316

Table I conveys the message that the approximated Koopman operators are able to fit the training data with acceptable accuracy overall. The RFI method produces the highest fit, and all HOD fits are comparable, with HOD having the best fit. It is also interesting to note that using nonlinear inputs in fact results in a slightly inferior fitting performance than using the linear inputs. To determine how the operators capture the full dynamics of the robot, their predictive performance is tested against other sets of data.

B. Comparison of Prediction Performance

To measure how well the approximate Koopman operators capture the dynamics of the soft robotic swimmer, we use the resulting operators to predict the evolution of the states for a time horizon $T = 15$ s, given a particular set of inputs u that were not used in the training, and then compare the resulting trajectories against the corresponding CFD data. In particular, four data sets that capture the soft robot's dynamics with different linearly varying inputs are considered. Fig. 4 illustrates the NMSE of each approach obtained for every data set when considering both linear and nonlinear inputs. Furthermore, for the sake of illustration, Fig. 5 illustrates the comparison for the RFI and HOD methods only in the case when nonlinear inputs are considered.

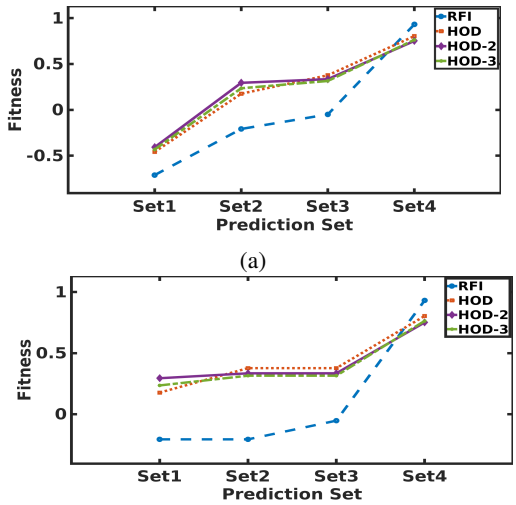
From Fig. 4, when considering the goodness of fit across different data sets, it is evident that the HOD methods, on average, achieve better prediction performance than the RFI-based method. In other words, although from Table I one can conclude RFI method produces the highest fit for the training data, from Fig. 4 one can see that when considering the predictive power of the trained Koopmans across different data sets, the HOD methods on average yield the highest fit. On the other hand, there seems to be no significant added benefit in including second and third-order derivatives when considering the fitness that is achieved. We conjecture that this is because the rigid body dynamics are not an accurate representation of the soft swimmer and therefore higher-order terms may not be relevant. Utilizing the first-order derivatives, however, does have the benefit of yielding a lower dimensional Koopman operator. Furthermore, it can be noted that there is a difference in the predictive performance between the Koopman operators trained using linear inputs or nonlinear inputs, where the former leads to higher level of fitness.



(a) RFI method.

(b) HOD method using 1st order derivatives

Fig. 3: Fitness between Koopman model and measurements used to train it for the two different approaches: (a) RFI and (b) HOD when nonlinear inputs are considered. The blue line shows the evolution of the states using the Koopman model.



(a)

(b)

Fig. 4: Level of fitness between four different data sets and Koopman predictions for all four methods when considering (a) nonlinear inputs and (b) linear inputs.

Upon further analysis of the Koopman operators, we note that although all four methods yield state transition A matrices with unstable eigenvalues, the resultant linear systems are either stabilizable or controllable. In particular, when considering both linear and nonlinear inputs, only the HOD method yields a system that is controllable.

We conclude that although the RFI method yields an operator that fits its training data the best, on average the HOD method provides the best fit when considering other data sets that capture different dynamics of the soft robotic swimmer. Furthermore, this method has the benefit of producing a lower dimensional operator and a resultant controllable linear system. These properties are important since lower dimensional linear representations reduce the complexity in synthesizing controllers, and the controllability plays a crucial role when addressing different control problems. In general, these results indicate that it is possible to produce a control-affine linear model that reasonably predicts the complex behavior of the soft swimmer without having any prior knowledge of the dynamics.

VI. Conclusions

In this paper, two different methods of obtaining basis functions for constructing Koopman operators are proposed for soft robotic swimmers. One method that utilizes higher-order derivatives of the states estimated using high-gain observers, and the other that utilizes the dynamic model of a rigid robotic fish to obtain the higher-order derivatives. The predictive performance of the estimated Koopman operators is tested against CFD simulation data and is quantified using the normalized squared error. It is shown that although both methods provide promising results, estimating higher-order derivatives from data appears to be more effective, and the resulting model is more computationally efficient for control design.

For future work, the Koopman operator obtained using the proposed methods will be used to design different model-based controllers for the soft robotic swimmer, which will be evaluated in CFD simulation. We will further prototype a soft robotic swimmer, and explore the use of the proposed modeling approach for such physical robots.

References

[1] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

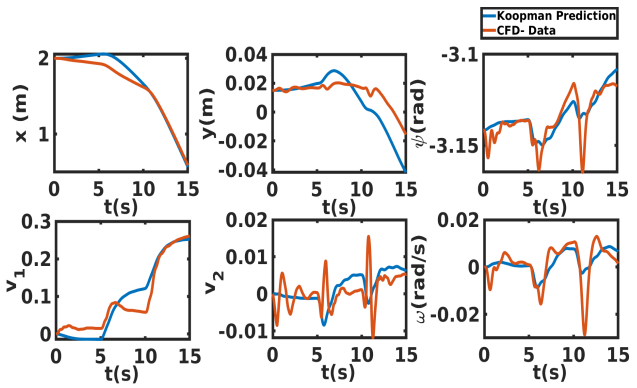
[2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[3] H.-T. Lin, G. G. Leisk, and B. Trimmer, "Goqbot: a caterpillar-inspired soft-bodied rolling robot," *Bioinspiration & biomimetics*, vol. 6, no. 2, p. 026007, 2011.

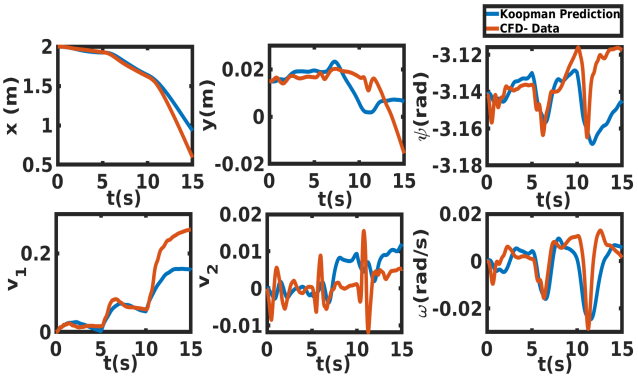
[4] M. Yamakita, N. Kamamichi, T. Kozuki, K. Asaka, and Z.-W. Luo, "A snake-like swimming robot using ipmc actuator and verification of doping effect," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2035–2040.

[5] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, "Review of fish swimming modes for aquatic locomotion," *IEEE Journal of oceanic engineering*, vol. 24, no. 2, pp. 237–252, 1999.

[6] M. L. Castaño, A. Mavrommati, T. D. Murphey, and X. Tan, "Trajectory planning and tracking of robotic fish using ergodic exploration," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 5476–5481.



(a) HOD Method using 1st order derivatives



(b) RFI Method

Fig. 5: HOD and RFI method fitness between Koopman model and simulation measurements for one actuation case when varying amplitude α_0 with constant bias β . The blue line shows the evolution of the states using the Koopman model.

[7] O. Ennasr, G. Mamakoukas, T. Murphey, and X. Tan, "Ergodic exploration for adaptive sampling of water columns using gliding robotic fish," in *ASME 2018 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2018.

[8] L. Wen, Z. Ren, V. Di Santo, K. Hu, T. Yuan, T. Wang, and G. V. Lauder, "Understanding fish linear acceleration using an undulatory biorobotic model with soft fluidic elastomer actuated morphing median fins," *Soft robotics*, vol. 5, no. 4, pp. 375–388, 2018.

[9] W.-S. Chu, K.-T. Lee, S.-H. Song, M.-W. Han, J.-Y. Lee, H.-S. Kim, M.-S. Kim, Y.-J. Park, K.-J. Cho, and S.-H. Ahn, "Review of biomimetic underwater robots using smart actuators," *International journal of precision engineering and manufacturing*, vol. 13, no. 7, pp. 1281–1292, 2012.

[10] H. Godaba, J. Li, Y. Wang, and J. Zhu, "A soft jellyfish robot driven by a dielectric elastomer actuator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 624–631, 2016.

[11] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, Y. Luo, *et al.*, "Fast-moving soft electronic fish," *Science Advances*, vol. 3, no. 4, p. e1602045, 2017.

[12] J. Shintake, V. Cacucciolo, H. Shea, and D. Floreano, "Soft biomimetic fish robot made of dielectric elastomer actuators," *Soft Robotics*, vol. 5, no. 4, pp. 466–474, 2018.

[13] A. D. Marchese, C. D. Onal, and D. Rus, "Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators," *Soft Robotics*, vol. 1, no. 1, pp. 75–87, 2014.

[14] O. Z. Jedidiah, "The design, modeling, and optimization of a biomimetic soft robot for fluid pumping and thrust generation using electroactive polymer actuators," Master's thesis, University of Nevada, Las vegas, 2018.

[15] S. K. Rajendran and F. Zhang, "Learning Based Speed Control of Soft Robotic Fish," in *Dynamic Systems and Control Conference*, vol. Volume 1, Atlanta, Georgia, USA, September 30 2018, v001T04A005.

[16] H. Hu, J. Liu, I. Dukes, and G. Francis, "Design of 3d swim patterns for autonomous robotic fish," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 2406–2411.

[17] Eunjeong Lee, "Design of a soft and autonomous biomimetic microrobotic fish," in *2010 5th IEEE Conference on Industrial Electronics and Applications*, June 2010, pp. 240–247.

[18] G. Mamakoukas, M. A. MacIver, and T. D. Murphey, "Feedback synthesis for underactuated systems using sequential second-order needle variations," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1826–1853, 2018.

[19] G. Mamakoukas, M. A. MacIver, and T. Murphey, "Sequential action control for models of underactuated underwater vehicles in a planar ideal fluid," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 4500–4506.

[20] A. Mauroy and J. Goncalves, "Linear identification of nonlinear systems: A lifting technique based on the Koopman operator," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6500–6505.

[21] M. O. Williams and C. W. Kevrekidis, Ioannis G.and Rowley, "A datadriven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, p. 13071346, Dec 2015.

[22] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Local Koopman operators for data-driven control of robotic systems," in *Robotics: Science and Systems XV*, Jun 2019.

[23] A. Mauroy and I. Mezic, "Global stability analysis using the eigenfunctions of the Koopman operator," *arXiv:1408.1379 [math]*, Sep 2015. [Online]. Available: http://arxiv.org/abs/1408.1379

[24] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 5, p. 315, 1931.

[25] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLOS ONE*, vol. 11, no. 2, p. e0150171, Feb 2016.

[26] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *arXiv:1707.01146 [math]*, Apr 2018. [Online]. Available: http://arxiv.org/abs/1707.01146

[27] S. P. Nandanoori, S. Sinha, and E. Yeung, "Data-driven operator theoretic methods for global phase space learning," *arXiv:1910.03011 [cs, eess,math]*, Oct 2019. [Online]. Available: http://arxiv.org/abs/1910.03011

[28] M. Haseli and J. Corts, "Efficient identification of linear evolutions in nonlinear vector fields: Koopman invariant subspaces," *arXiv:1909.01419 [cs, eess, math]*, Sep 2019. [Online]. Available: http://arxiv.org/abs/1909.01419

[29] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition," in *Advances in Neural Information Processing Systems*, 2017, pp. 1130–1140.

[30] M. Haseli and J. Cortés, "Parallel learning of koopman eigenfunctions and invariant subspaces for accurate long-term prediction," *arXiv preprint arXiv:2005.06138*, 2020.

[31] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, p. 909930, Jan 2018.

[32] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the Koopman operator and model predictive control," *arXiv:1902.02827 [cs]*, Jul 2019. [Online]. Available: http://arxiv.org/abs/1902.02827

[33] K. Hara, M. Inoue, and N. Sebe, "Learning koopman operator under dissipativity constraints," *arXiv preprint arXiv:1911.03884*, 2019.

[34] G. Mamakoukas, I. Abraham, and T. D. Murphey, "Learning data-driven stable koopman operators," *arXiv preprint arXiv:2005.04291*, 2020.

[35] A. Hess, X. Tan, and T. Gao, "CFD-based multi-objective controller optimization for soft robotic fish with muscle-like actuation," *Bioinspiration & Biomimetics*, 2020.

[36] J. Wang and X. Tan, "Averaging tail-actuated robotic fish dynamics through force and moment scaling," *IEEE Transactions on Robotics*, vol. 31, no. 4, p. 906917, Aug 2015.

[37] H. K. Khalil and L. Praly, "High-gain observers in nonlinear feedback control," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 6, pp. 993–1015, 2014.