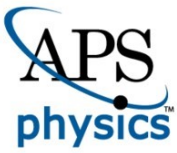

Learning to swim in potential flow

Yusheng Jiao, Feng Ling, Sina Heydari, Nicolas Heess, Josh Merel, and Eva Kanso

Phys. Rev. Fluids **6**, 050505 — Published 12 May 2021

DOI: 10.1103/PhysRevFluids.6.050505

Learning to swim in potential flow

Yusheng Jiao, Feng Ling, Sina Heydari, and Eva Kanso[1]
Department of Aerospace and Mechanical Engineering,
University of Southern California, Los Angeles, CA 90089

Nicolas Heess and Josh Merel
DeepMind, London

Fish swim by undulating their bodies. These propulsive motions require coordinated shape changes of a body that interacts with its fluid environment, but the specific shape coordination that leads to robust turning and swimming motions remains unclear. To address the problem of underwater motion planning, we propose a simple model of a three-link fish swimming in a potential flow environment and we use model-free reinforcement learning for shape control. We arrive at optimal shape changes for two swimming tasks: swimming in a desired direction and swimming towards a known target. This fish model belongs to a class of problems in geometric mechanics, known as *driftless* dynamical systems, which allow us to analyze the swimming behavior in terms of geometric phases over the shape space of the fish. These geometric methods are less intuitive in the presence of *drift*. Here, we use the shape space analysis as a tool for assessing, visualizing, and interpreting the control policies obtained via reinforcement learning in the absence of drift. We then examine the robustness of these policies to drift-related perturbations. Although the fish has no direct control over the drift itself, it learns to take advantage of the presence of moderate drift to reach its target.

Keywords: Fish swimming, Reinforcement learning, Sensorimotor control.

I. INTRODUCTION

Fish swim through interactions of body deformations with the fluid environment. A fish assimilates sensory information about its own body and the external environment and produces patterns of muscle activation that result in desired body deformations; see Fig. 1A. How these sensorimotor decisions are enacted at the physiological level, at the level of neuronal circuits, remains unclear [1–4]. Animal models, such as the *Danio rerio* zebrafish [5–7], as well as robotic and mathematical models [8, 9], provide valuable insights into the sensorimotor control underlying fish behavior. Such understanding offers enticing paradigms for the design of artificial soft robotic systems in which the control is *embodied* in the physics [10, 11]. *Embodied* systems sense and respond to their environment through a physical body [12, 13]; physical interactions with the environment are thus vital for sensing and control. In fish, the dynamics of the fluid environment is essential both at an evolutionary time scale – in shaping body morphologies [14] and sensorimotor modalities – and at a behavioral time scale. Fish bodies are tuned to exploit flows [15, 16]. Body designs and undulatory motions have been examined in computational and semi-analytical models of fluid-structure interactions [17–21], including models of body stiffness and neuromechanical control [22, 23]. The fish's ability to integrate multiple sensory modalities such as vision [24–26] and flow sensing [27, 28] are essential to behaviors ranging from rheotaxis [29–31] to schooling [32–35]. Recent developments prove that machine learning techniques are highly effective in addressing problems of flow sensing and fish behavior [36–43].

A central problem in fish behavior, which is also relevant for underwater robotic systems, is *gait design* or *motion planning*: what body deformations produce a desired swimming objective? The answer requires an understanding of how the numerous biomechanical degrees of freedom of the fish body are coordinated to achieve the common objective. Mathematically, this problem is often expressed in terms of an optimality principle: find control laws that optimize a desired objective, such as maximizing swimming speed or minimizing energetic cost [17, 18, 44]. But how these control laws are implemented in the nervous system, and how they are acquired via learning algorithms, are typically beyond the scope of such methods. As these optimization methods rely on an internal model of the dynamics, different computational results have been obtained by varying the specification of the physical model of the fish, the performance metric, and the control constraints [17, 18, 44, 45].

Model-free reinforcement learning (RL) of embodied systems offers an alternative framework for gait design that is mathematically and computationally tractable [47–49]. In this framework, the fish's world is divided into a body

[1] kanso@usc.edu; https://sites.usc.edu/kansolab/

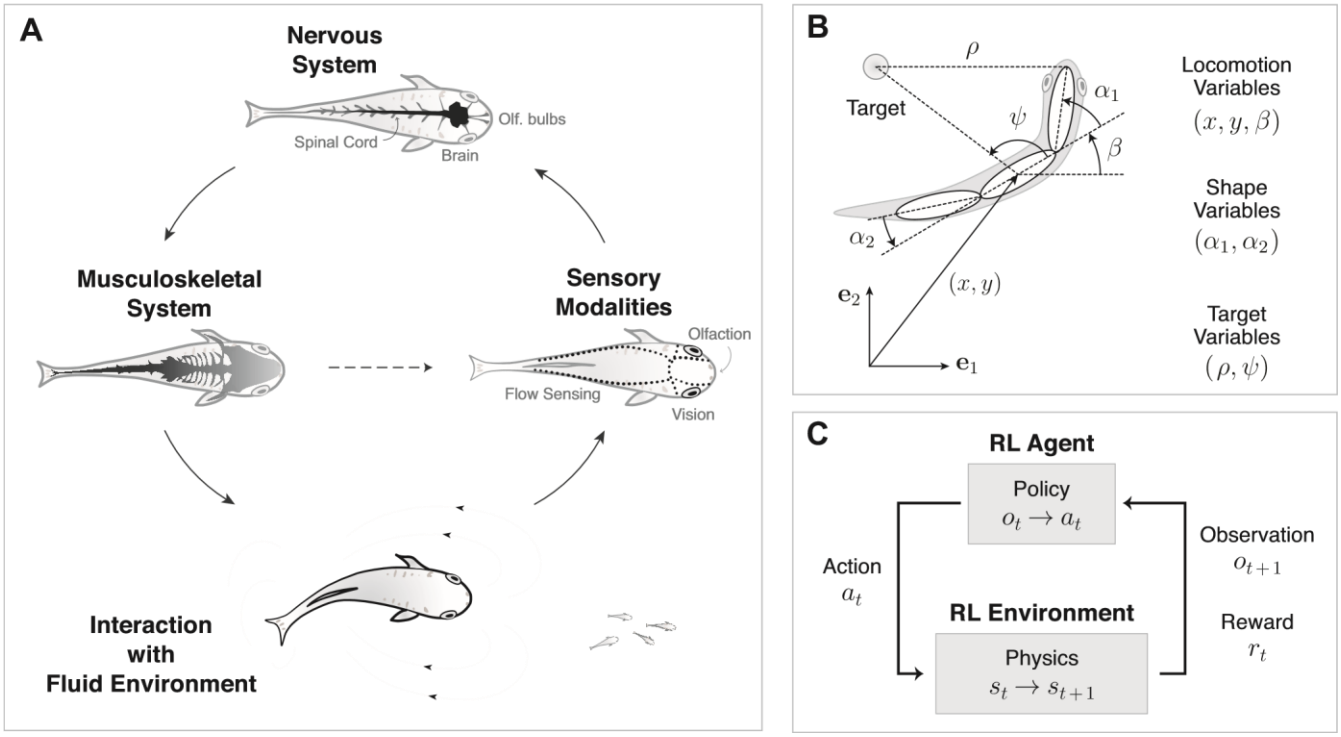

Figure 1. **Model-free reinforcement learning and the three-link fish.** A. Illustration of sensorimotor feedback loops in fish. Motor commands generated in the nervous system activate the musculoskeletal system, resulting in deformations of the body. Body deformations, through interaction with the fluid environment, lead to swimming meanwhile sensory modalities provide sensory feedback to the nervous system. The dashed arrow between musculoskeletal and sensory systems indicates somatic sensing used to assess whether previous motor commands were successfully executed. Other reflexive or preflex signals could also be at play [2, 27, 46]. B. Three-link fish swimming in quiescent fluid. Locomotion variables (x,y,β) are set in a lab fixed frame, while the shape variables (α_1,α_2) and target variables (ρ,ψ) are set in body frame symbolizing egocentric control and learning. C. To apply model-free reinforcement learning to our problem, we only need to set the appropriate state, observation, action, and reward variables based on our fish model.

controlled by a *learning agent* and an environment that encompasses everything outside of what the agent can control. The agent can be viewed as an abstract representation of the parts of the fish responsible for sensorimotor decisions. In RL, the agent must learn from experience in a trial-and-error fashion. Specifically, repeated interactions of the body with the environment enable the agent to collect *sensory observations*, *control actions*, and associated *rewards*. The goal of the agent is thus to learn to produce behavior that maximizes rewards, and the process is model-free when learning does not make use of either a priori or developed knowledge of the physics of the system. The learned feedback control law, called a *policy*, is essentially a mapping from sensory observations to control actions. This mapping is nonlinear and stochastic, and, by construction, rather than providing a single optimal trajectory, it can be applied to any initial condition and transferred to conditions other than those seen during training such as when the body or fluid environment are perturbed [50].

Here, we employ RL to design swimming gaits. We use an idealization in which the fish is modeled as an articulated body consisting of three links, with front and rear links free to rotate relative to the middle link via hinge joints [44, 45, 51–54]. In describing the physics of the fish, we cede the complexity of accounting for the full details of the fluid medium in favor of considering momentum exchange between the articulated body and the surrounding fluid in the context of a potential flow model [44, 45, 52, 53]. This model is a canonical example of a class of under-actuated control problems whose dynamics can be described over the actuation (shape) space using tools from geometric mechanics [44, 55–58]. Specifically, swimming motions can be represented by the sum of a *dynamic phase* or *drift*, and a *geometric phase* over the shape space of all body deformations [45]. In the geometric phase case, a geometric *connection*, defined as vector fields over the actuation shape space, maps infinitesimal shape changes into infinitesimal rigid motions of the whole body. From a motion control perspective, this geometric framework is advantageous in that it provides tools for gait analysis and manual design of control policies over the full shape space [54]. These geometric tools also provide an

intuitive way for direct and interpretable visualizations of the RL-based policies. Here, we visualize the RL policies as vector fields over the shape space. This framework leads to a straightforward interpretation of the RL policy: given

an observation of its own shape, the fish's action needs to follow the corresponding vector in the shape space. A trajectory or realization of the RL policy arises from locally following these vectors to achieve a desired global task. These tools also allow us to probe the optimality and behavior of the RL policy in light of the physics of the system and in comparison to manually-designed policies over the fish shape space.

II. MATHEMATICAL MODEL OF THE THREE-LINK FISH

Consider a three-link fish as shown in Fig. 1(B). Rotations of the front and rear links relative to the middle link are denoted by the angles α_1 and α_2 such that (α_1,α_2) fully describe all possible body deformations. We constrain the swimming motions to a two-dimensional plane, and let (x,y) and β denote the net planar displacement and rotation of the fish body, such that $(\dot{x},\dot{y})$ and $\dot{\beta}$ represent the linear and rotational velocities of the fish in inertial frame (the dot notation $\dot{()} = d()/dt$ represents differentiation with respect to time t). We also introduce the linear velocity (v_1,v_2) expressed in a co-rotating body frame attached to the center of the middle link,

$$v_1 = \dot{x}\cos\beta + \dot{y}\sin\beta, \qquad v_2 = -\dot{x}\sin\beta + \dot{y}\cos\beta. \tag{1}$$

The total linear momentum (p_x,p_y) and total angular momentum π of the body-fluid system are expressed in the inertial frame, and they are related to their counterparts (P_1,P_2) and Π in body-fixed frame as follows,

$$p_x = P_1\cos\beta - P_2\sin\beta, \qquad p_y = P_1\sin\beta + P_2\cos\beta, \qquad \pi = \Pi. \tag{2}$$

In potential flow, it is a known result that the total linear and angular momenta of the body-fluid system can be expressed in terms of the body geometry and velocity, via the so-called *added mass* matrices [44, 45, 59]. Expressions for the added mass matrices of the three-link fish when the links are hydrodynamically decoupled (coupled only geometrically through the holonomic constraints at the joints) are derived in detail in Appendix A. The total momenta (P_1,P_2) and Π are given by

$$\begin{bmatrix} P_1 \\ P_2 \\ \Pi \end{bmatrix} = \mathbb{I}_{\text{lock}} \begin{bmatrix} v_1 \\ v_2 \\ \dot{\beta} \end{bmatrix} + \mathbb{I} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} \tag{3}$$

P v couple,

where $\mathbb{I}_{\text{lock}}$ is the locked mass matrix at a given shape of the fish (see Eqs. A6-A9) and $\mathbb{I}_{\text{couple}}$ is the mass matrix associated with shape deformations (see Eq. A10).

In the absence of external forces and moments on the fish-fluid system, the total momenta (p_x,p_y) and π are conserved for all time. Conservation of total momentum yields, upon inverting (2) and substituting into (3),

$$\begin{bmatrix} v_1 \\ v_2 \\ \dot{\beta} \end{bmatrix} = \mathbb{I}_{\text{lock}}^{-1} \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ \pi \end{bmatrix} - \mathbb{I}_{\text{lock}}^{-1}\mathbb{I} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix}$$

v couple. (4)

If we further substitute (1) into (4), we arrive at three coupled first-order equations of motion for x, y, β given α_1 and α_2. The control problem consists of finding the time evolution of shape changes $(\alpha_1(t),\alpha_2(t))$ that achieve a desired locomotion task $(x(t),y(t))$ and $\beta(t)$. Specifically, a swimming gait is defined as a cyclic shape change $(\alpha_1(t),\alpha_2(t))$, with period T, that results in a net swimming $(x(t),y(t))$ or turning $\beta(t)$ of the fish body.

This model is a canonical example of a class of under-actuated control problems whose dynamics can be described over the *shape space* using tools from geometric mechanics [44, 55–57]. On the right-hand side of (4), the first term represents a *dynamic phase* or *drift* and the second term represents a *geometric phase* over the fish shape space (α_1,α_2) [45]. The geometric phase is best described in terms of the *local connection* matrix A [45, 54], which is a function only of the shape variables α_1 and α_2,

$A_{11}\,A_{12}$

$$\mathbb{A} = \begin{bmatrix} A_{21} & A_{22} \\ A_{\beta 1} & A_{\beta 2} \end{bmatrix} := -\mathbb{I}_{\text{lock}}^{-1}\mathbb{I} \quad \text{couple.} \tag{5}$$

Each row of A describes a nonlinear vector field over the shape space, giving rise to three vector fields $\mathbf{A}_1 \equiv (A_{11},A_{12})$, $\mathbf{A}_2 \equiv (A_{21},A_{22})$, and $\mathbf{A}_\beta \equiv (A_{\beta 1},A_{\beta 2})$ over the (α_1,α_2) plane as shown in Fig. 2(A). In driftless systems, net locomotion is fully controlled by the fish shape changes as dictated by the connection matrix A. However, in the presence of drift, shape control is not sufficient to determine the fish motion in the physical space, which is then affected by the drift term in (4).

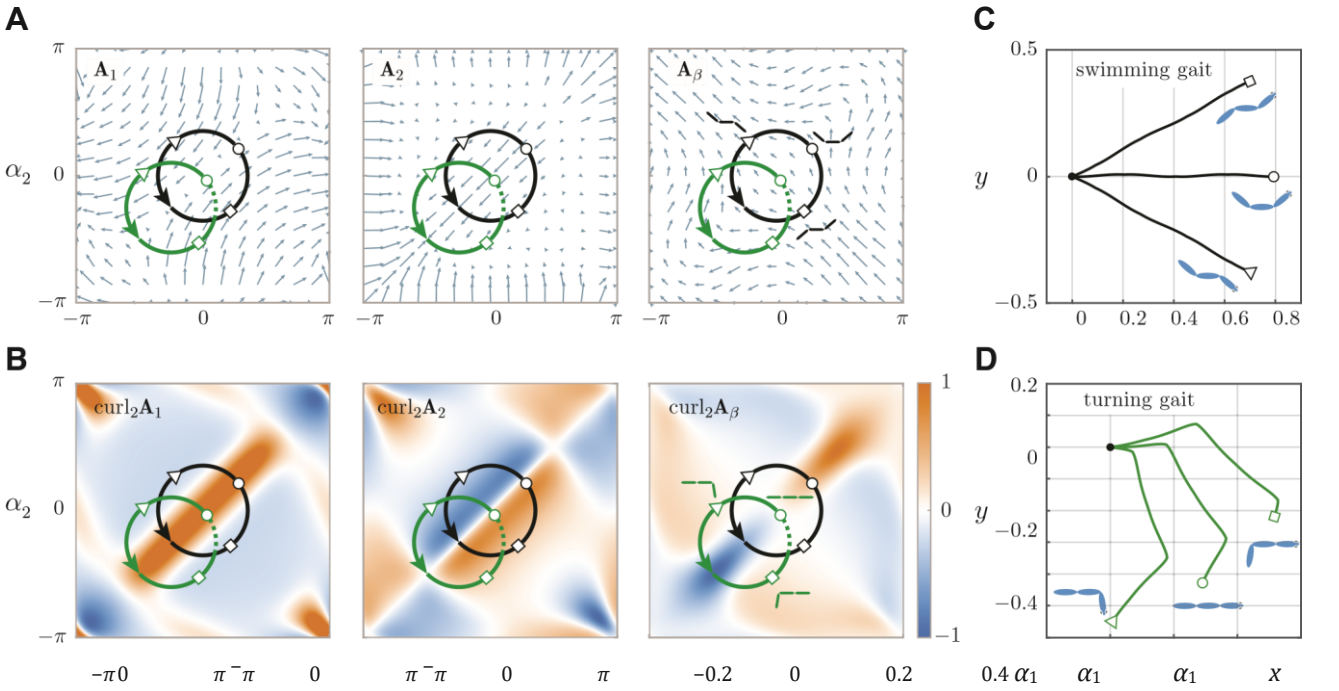

Figure 2. **Using connection matrix for simple gait design.** A and B. Rows of connection matrix A give us three vector fields, $\mathbf{A}_1 \equiv \partial v_1/\partial \alpha_i$, $\mathbf{A}_2 \equiv \partial v_2/\partial \alpha_i$, and $\mathbf{A}_\beta \equiv \partial \beta/\partial \alpha_i$, the curl of which yield three corresponding scalar fields. Magnitude of the scalar fields are normalized to be within [−1,1]. Value of the scalar fields can facilitate the design of simple swimming and turning gaits, as shown above by black and green circles respectively. Six body configurations, each corresponding to points marked by black and green °, △, , are sketched for additional clarity. C and D. Fish that start with body centered at the origin of the x-y plane and follow the same gait circle swim/turn in different directions in the physical space depending on initial body shapes; the initial shape is depicted in blue at the end of each trajectory.

III. GEOMETRIC PHASES

Geometric phases are defined as the net locomotion (x,y,β) that results from prescribed cyclic shape changes in the (α_1,α_2) plane at zero total momentum (no drift). Inverting (1) and substituting (5) into (4) at zero total momentum, we arrive at

$$\begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{\beta 1} & A_{\beta 2} \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix}. \tag{6}$$

Motions (x,y,β) in the physical space are obtained by integrating (6) with respect to time. Rotations are directly proportional to the line integral of the vector field $\mathbf{A}_\beta$ as evident by integrating the last equation in (6),

$$\beta(T) - \beta(0) = \mathrm{I}\, \mathrm{d}\beta = \mathrm{I}\, (A_{\beta 1}\mathrm{d}\alpha_1 + A_{\beta 2}\mathrm{d}\alpha_2). \tag{7}$$

c c

Here, T is the time-period for going around the closed trajectory in the shape space once. Using Green's Theorem, we get

$$\beta(T) - \beta(0) = \iint \left(\frac{\partial A_{\beta 2}}{\partial \alpha_1} - \frac{\partial A_{\beta 1}}{\partial \alpha_2} \right) \mathrm{d}\alpha_1 \mathrm{d}\alpha_2 = \iint \mathrm{curl}_2(\mathbf{A}_\beta) \mathrm{d}\alpha_1 \mathrm{d}\alpha_2 \,. \tag{8}$$

Here, curl_2 denotes the two-dimensional curl as a scalar field over the (α_1,α_2) plane. The scalar field $\mathrm{curl}_2(\mathbf{A}_\beta)$ provides an intuitive tool for understanding the effect of a cyclic shape deformation on the net rotation of the fish: net rotations are proportional to the integral of $\mathrm{curl}_2(\mathbf{A}_\beta)$ over the area enclosed by a closed shape trajectory; see Fig. 2. However, translational motions (x,y) are not directly proportional to the area integrals of $\mathrm{curl}_2(\mathbf{A}_1)$ and $\mathrm{curl}_2(\mathbf{A}_2)$, but to a combination of all three integrals coupled through the fish rotational dynamics as evident from (6). This coupling is clear when the equations of motion for x and y in (6) are written in scalar form,

$$\begin{aligned} \dot{x} &= [A_{11}\dot{\alpha}_1 + A_{12}\dot{\alpha}_2]\cos\beta - [A_{21}\dot{\alpha}_1 + A_{22}\dot{\alpha}_2]\sin\beta, \\ \dot{y} &= [A_{11}\dot{\alpha}_1 + A_{12}\dot{\alpha}_2]\sin\beta + [A_{21}\dot{\alpha}_1 + A_{22}\dot{\alpha}_2]\cos\beta. \end{aligned} \tag{9}$$

Despite this complication, the scalar fields defined by $\mathrm{curl}_2(\mathbf{A}_1)$, $\mathrm{curl}_2(\mathbf{A}_2)$, and $\mathrm{curl}_2(\mathbf{A}_\beta)$, shown in Fig. 2(B), are informative of the net translational (x,y) and rotational β motions of the fish. To illustrate the utility of these curl fields, we show two examples of cyclic shape changes depicted in black and green lines. A fish changing its shape following the black line undergoes zero net rotation because the area integral of $\mathrm{curl}_2(\mathbf{A}_\beta)$ is identically zero, but it swims forward in the (x,y) plane. The net displacement per period is a conserved quantity, whereas the direction of motion depends on a combination of the fish initial shape $(\alpha_1(0),\alpha_2(0))$ and initial orientation $\beta(0)$ as shown in Fig. 2(C). Here, we consider three different initial shapes, depicted by the markers ∘, △, , all at $\beta(0) = 0$. Similarly, shape deformations following the green line lead to net reorientations in the physical space, as shown in Fig. 2(D). Evidently, no net motion occurs if the shape trajectory is degenerate, that is, does not enclose an area in the shape space. Further a re-scaling of time does not affect the net motion of the fish, only the speed at which the fish completes these cyclic shape changes.

In Fig. 2 and hereafter, the equations of motion are non-dimensionalized using the total length of the fish and the total mass in the head-to-tail direction of the straight fish as the characteristic length and mass scale. Specifically, we set the total mass of the fish body to be equal to the added mass (actual mass of the fish is negligible). We leave the time scale unchanged because the characteristic time depends on the speed of shape changes, which is a control parameter to be determined by the controller.

The scalar fields $\mathrm{curl}_2(\mathbf{A}_1)$, $\mathrm{curl}_2(\mathbf{A}_2)$, $\mathrm{curl}_2(\mathbf{A}_\beta)$ over the shape space encode information about the net locomotion of the fish in a driftless environment, and can be used to design simple swimming and turning gaits as shown in Fig. 2. However, these geometric tools do not allow for a straightforward design of control policies for arbitrary motion planning [52, 54], and they are even less instructive in the presence of drift. Next we consider an RL driven approach for motion control.

IV. MOTION CONTROL VIA REINFORCEMENT LEARNING

We use RL to train the three-link fish on two different tasks: (i) to swim parallel to a desired direction in a driftless environment; (ii) to swim towards a target point located at a distance ρ and angular position ψ from the fish nose, with ρ and ψ expressed in the fish frame of reference (Fig. 1B). Given the rotational symmetry of the fish-fluid space, in the first task, we fix the desired direction to be parallel to the x-axis without loss of generality. For the second task, we first train the fish in a driftless environment, then introduce drift and train again in the presence of drift. The first task allows for direct comparison of the performance of the trained policy to manually-designed policies in the context of geometric mechanics as described in § III. The second task allows for evaluation and comparison of the performance of the driftless and drift-aware policies under environmental perturbations.

Central to any RL implementation are the notions of the *state* of the system, the *observations* given to the learning agent, the *actions* taken by the agent, and the *rewards* given to the agent in light of its behavior. The state s_t of the fish-fluid-target system at a time t is given by the fish position and orientation in inertial frame (x,y,β), its shape (α_1,α_2), and the target position relative to the fish (ρ,ψ); see Fig. 1(B,C). As sensory input, we provide the fish a set of observations based on its proprioception of its own shape α_1 and α_2, as well as an egocentric observation of the task, namely for controlling the direction of swimming, the fish knows the desired swimming direction relative to itself $\psi = -\beta$, and for

swimming towards a target, it knows the relative angular position ψ of the target point. This yields a set of observations $o_t = (\alpha_1,\alpha_2,\psi)_t$. Additionally, when training in the presence of drift, the magnitude and direction of the drift vector (p_x,p_y) are also provided as observations. As control action, the fish has direct control of its shape using the rate of shape changes as actions $a_t = (\dot{\alpha}_1,\dot{\alpha}_2)_t$. With this choice of action, the control can be projected onto the shape space and directly compared to the geometric mechanics approach. We constrain the value of the actions to be between −1 and 1 rad per unit time, and we impose limits on the joint-angle so that α_1 and α_2 are only allowed to change between $-2\pi/3$ and $2\pi/3$ rad to avoid self-intersection.

In RL, the decision making process is modeled as a stochastic control policy $\pi_\theta(a_t|o_t)$ that produces actions a_t given observations o_t of the state of the fish-fluid system. The policy is parameterized by a set of parameters θ to be optimized. An optimal policy is learned to produce behavior that maximizes rewards. We use a dense shaping reward, that is, the fish is given a reward at every decision time step. Specifically, we set the reward to be the distance the fish travels in the desired direction towards the target state. For learning to swim parallel to the x-axis, we use the reward $r_t = x_{\text{nose},t+1} - x_{\text{nose},t}$, which is the change in the fish nose position along the x-axis. Note that this choice of reward function breaks the head-tail symmetry of the fish favoring higher reward when turning towards the positive x-axis. For learning to swim towards a target, the reward $r_t = \rho_t - \rho_{t+1}$ is based on the change in the relative distance ρ from the fish nose to the target. The *return* $R_t = \sum_{t'}^{\infty} \gamma^{t'-t} r_{t'}$ is defined as the infinite horizon objective based on the sum of discounted future rewards, where $\gamma \in [0,1]$ is known as the discount factor; it determines the preference for immediate over future rewards. We set $\gamma = 0.99$ to make the fish foresighted. The goal is to arrive at an optimal set of parameters θ that maximizes the *expected* return $\mathcal{J}(\pi_\theta) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ for a distribution of initial states. Here, the expectation is taken with respect to the distribution over trajectories $\pi(\tau)$ induced jointly by the fish dynamics, viewed as a partially-observable Markov decision process, and the policy $\pi_\theta(a_t|o_t)$. One approach to solving this optimization problem is to use a policy gradient method that computes an estimate of the gradient $\nabla_\theta J$ for learning. Policy gradient methods are widely used to learn complex control tasks and are often regarded as the most effective reinforcement learning techniques, especially for robotics applications [60–64]. Here, we use a specific class of policy gradient methods, known as *actor-critic* methods [65, 66] where the fish learns simultaneously a policy (actor) and a value function (critic). We implement this method using the clipped advantage Proximal Policy Optimization (PPO) algorithm proposed in [67]. This algorithm ensures fast learning and robust performance by limiting the amount of change allowed for the policy within one update. A pseudo-code implementation of the PPO algorithm and additional implementation details are provided in Appendix B.

V. TRAINING THE FISH TO SWIM

We trained the three-link fish to (i) swim parallel to the x-direction in a driftless environment; (ii) swim towards a target point in the absence and presence of drift. We refer to the first task as *direction-control* for short, and the second task as *naive* and *drift-aware target-seeking*, respectively, based on their awareness of drift.

For the purpose of efficient training we imposed a finite time interval, following which the system state was reset to the initial state for a new round of training. Each round is referred to as an *episode*. In all training episodes, we initialized the fish center to be at the origin of the inertial frame, and we initialized the shape angles α_1,α_2 and body orientation β by sampling from a uniform distribution over all permissible angles to maximize the chances for robust learning. We fixed the maximum episode length to 150 time steps, with no early termination allowed. In training the target-seeking policies, the target was initially placed at a fixed distance (three-unit length) from the fish center but at a random orientation. For the drift-aware policy, a drift was introduced in the form of a non-zero total linear momenta p_x,p_y in (4). The drift magnitude was sampled from a uniform distribution between 0 and 0.15 and its direction from a uniform distribution between 0 and 2π, such that the drift was kept constant within a training episode. Based on the non-dimensional scales introduced in section III, one unit of drift aligned parallel to a straight fish causes it to translate one-unit length per unit-time.

For the direction-control policy, we performed 24 runs of RL training with 10,000 episodes in each run. The training process is illustrated in terms of reward evolution, calculated by taking the sum of all rewards in an episode, in Appendix B, Fig. B.1(A). There was some variability among the seeds, but most trained policies performed very well; only a single policy did not converge by the end of training episodes. Note that fluctuations in reward after policy convergence are partly due to the stochasticity built into the policy itself and partly due to variation in task difficulty given the random

initial conditions: different initial conditions require different amount of time and effort for the fish body to align with the *x*-axis.

It is worth pointing out here the training results of the direction-control policy were affected significantly by the episode length. In order to swim in a desired direction starting from an arbitrary initial orientation, the fish has to first turn in that direction, then swim forward. Policies trained with longer episodes performed better in the swimming portion of the trajectory but failed to make large-angle turns, as training data collected on swimming significantly outweighed those collected on turning. On the other hand, policies trained with shorter episodes made turns of any angle, but were less likely to swim straight after turning. We chose the episode length, following several trial and error trainings, to be 150 as a reasonable compromise to learn to both turn and swim effectively.

The evolution of rewards during training of the two target-seeking policies are plotted in Fig. B.1(B), each with 20 runs of the RL algorithm and 15,000 episodes in each run. The naive policy converged faster than the drift-aware policy, and both policies converged slower than the direction-control policy. These results indicate that the task itself, as well as variations in the environment and number of observations affect the convergence rate, that is, the learning difficulty. Note that the numerical value of the reward is not directly comparable between policies for different tasks.

We evaluated the performance of the trained policies by testing them under two type of conditions: conditions similar to those seen during training and perturbed conditions not seen during training, as discussed next.

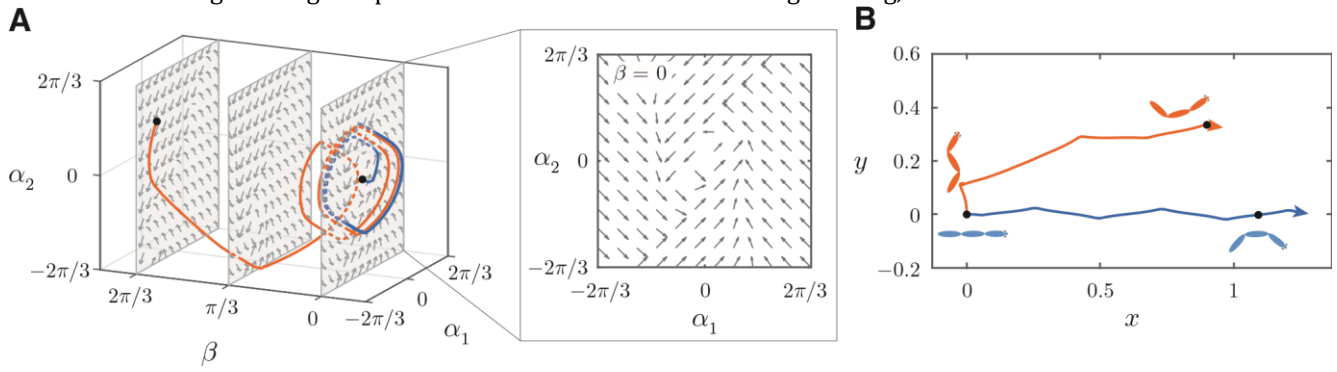

Figure 3. **Visualizing the direction-control RL policy.** A. Given the direction-control task, we visualize the mean RL policy actions ($\dot{\alpha}_1,\dot{\alpha}_2$) as vector fields in the observation space of (α_1,α_2,β). Two example observation trajectories starting at $\beta(0) = 0,\alpha_1(0) = 0,\alpha_2(0) = 0$ (blue) and $\beta(0) = 2\pi/3,\alpha_1(0) = -\pi/3,\alpha_2(0) = \pi/3$ (orange) are plotted with slices of the mean policy at $\beta = 0,\pi/3,2\pi/3$. The inset shows a flattened view of the slice at $\beta = 0$. B. Physical space trajectories of the corresponding examples shown in A are plotted together with the fish body configurations at the starting point and chosen points near the ends.

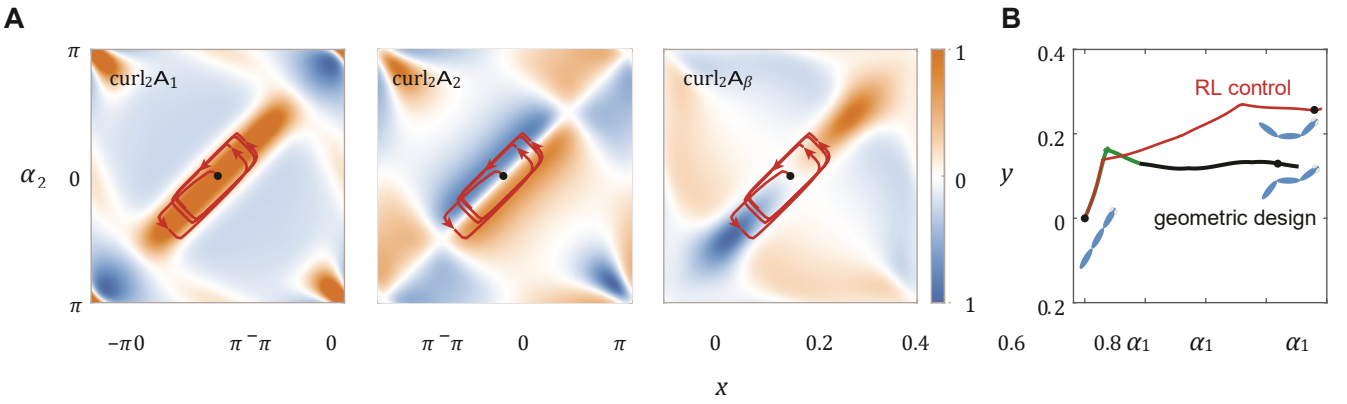

Figure 4. **RL provides smooth transition between turning and swimming gaits.** A. The shape space trajectory of the fish reorienting itself to swim parallel to the *x*-axis produced by mean RL policy is superimposed to the scalar curl fields from Fig. 2(B). Note that this trajectory starts off-centered and smoothly moves to cycles that are symmetric about the origin. B. The physical space trajectory due to the mean RL policy (red) in comparison to a manually patched turning-to-swimming trajectory (green and black) using circular gaits in Fig. 2. Without further fine-tuning on the shape of the gaits, the manually patched result shows a more abrupt and unnecessary turn angle. In both simulations, the fish start in a straight configuration with an orientation of $\beta(0) = \pi/3$ at the origin.

VI. BEHAVIOR OF TRAINED FISH

To visualize the RL direction-control policy, we plot in Fig. 3(A) the action vector fields $(\dot{\alpha}_1,\dot{\alpha}_2)$ over the observations space $(\alpha_1,\alpha_2,\beta)$. These vector fields depend on the orientation β of the fish in the physical space such that the control policy $(\dot{\alpha}_1,\dot{\alpha}_2)$ forms a *foliation* over β. Three slices of this foliation are highlighted. The right hand side of Fig. 3(A) provides a closer look at the policy slice at $\beta = 0$; the arrows indicate the mean actions advised by the policy for a given set of observations α_1,α_2 at $\beta = 0$. In Fig. 3(B), we show the details of two trajectories in the physical space starting from two distinct configurations. In the first test, the fish starts at zero orientation, $\beta(0) = 0$, in a straight shape, $(\alpha_1(0),\alpha_2(0)) = (0,0)$. The goal of the fish is simply to swim forward. In the second test, the initial orientation and shape are $\beta(0) = 2\pi/3$ and $(\alpha_1(0),\alpha_2(0)) = (-\pi/3,\pi/3)$, from which the fish needed to turn and swim along the x-axis. In both cases, the fish is able to turn to the desired direction and swim steadily. In Fig. 3(A), we highlight the corresponding trajectories in the $(\alpha_1,\alpha_2,\beta)$ space. As the fish moves through the physical space, β changes causing the fish to take actions from distinct slices of the foliation of action vector fields. Both trajectories tend to the same periodic swimming cycle around $\beta = 0$, indicating that the control actions are similar once both fish are aligned with the desired swimming direction.

We further explore the shape changes undertaken by the second (red) fish by superimposing these shape changes onto the scalar fields scalar fields $\mathrm{curl}_2\mathbf{A}_1$, $\mathrm{curl}_2\mathbf{A}_2$ and $\mathrm{curl}_2\mathbf{A}_\beta$ introduced in Section III; see Fig. 4(A). The corresponding

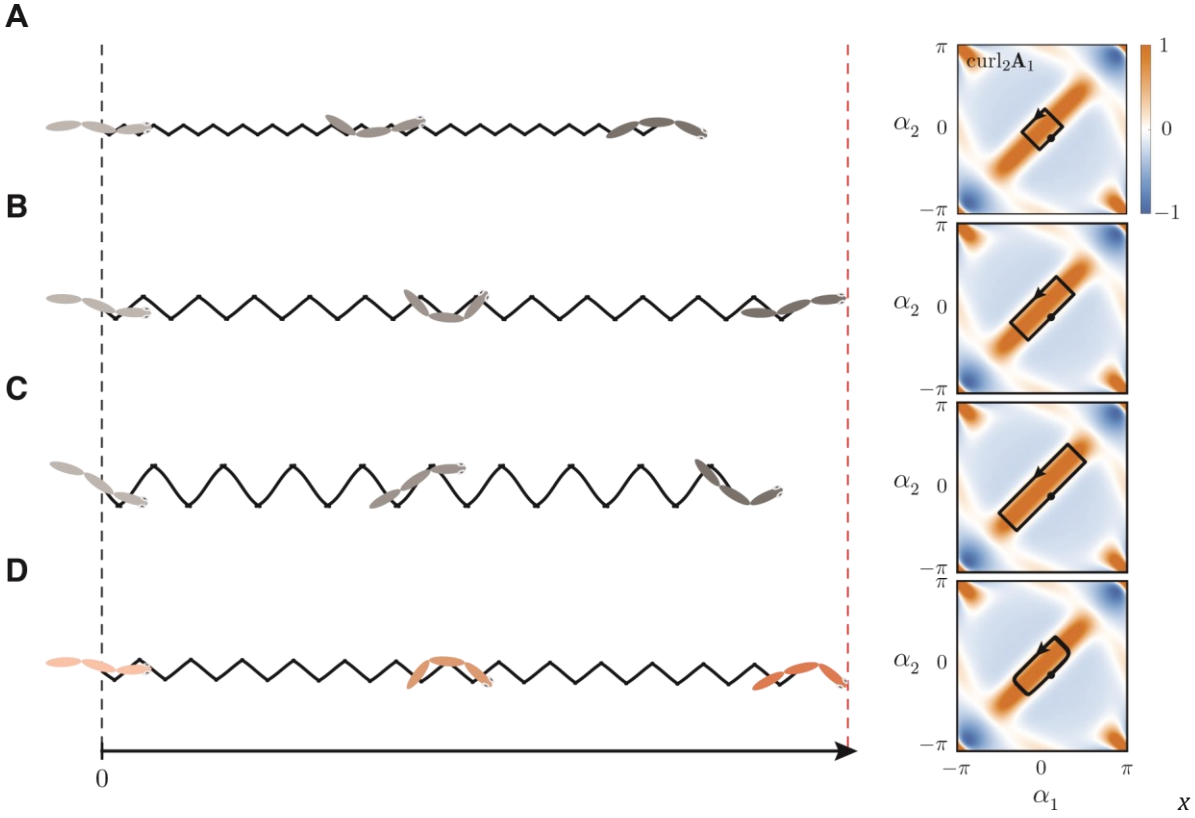

Figure 5. **Racing against the RL fish.** To test the optimality of our direction-control RL policy, we compare forward swimming performance between the geometrically-designed gaits and RL results. In A-C we use rectangular gaits with length equal to 1.2 (small), 2.4 (medium), and 3.6 (large), respectively, and width all equivalent to shape space trajectory following the mean actions from RL policy in D. All shape space trajectories are shown superimposed on top of the $\mathrm{curl}_2\mathbf{A}_1$ field on the right. Physical space trajectories on the left show that the mean RL policy achieves its excellent performance by choosing an optimal amount of lateral oscillation during forward swimming, while the small and large rectangular gaits move slower due to either insufficient or overwhelming side-way motion. Note that fish in A-C are initialized with the same shape but slightly different initial orientations to ensure they all swim in exactly the x-direction. In addition, all fish utilize the maximum actions allowed at each time step.

motion in the physical space is depicted in red in Fig. 4(B). The shape deformations produced by the RL policy can be interpreted as consisting of two regimes: an initial turning regime followed by a forward swimming regime. The turning regime is indicated by the initial portion of the shape trajectory enclosing most of the blue area in the $\mathrm{curl}_2\mathbf{A}_\beta$ image; the

integral in (7) along this portion of the trajectory is negative, leading to a clockwise rotation. The swimming regime is indicated by the periodic shape changes enclosing the rectangular orange portion in the $\text{curl}_2\mathbf{A}_1$ image. The area integral of $\text{curl}_2\mathbf{A}_1$ along this portion of the trajectory is positive, whereas the corresponding area integrals of $\text{curl}_2\mathbf{A}_2$ and $\text{curl}_2\mathbf{A}_\beta$ are identically zero, leading to net motion in the positive x-direction. These shape deformations and resulting turning and swimming motions can be compared to a manually-designed shape trajectory based on the turning (green) and swimming (black) gaits in Fig. 2(A,B). Specifically, starting from a straight fish configuration, we follow the solid portion of the green trajectory (turning) in Fig. 2(A,B), and transition into the black trajectory (swimming) at the second intersection of the green and black shape trajectories. The resulting motion in the physical space is superimposed onto Fig. 4(B). Both the RL and manually patched gaits lead the fish to turn and swim parallel to the x-axis, however, in the RL produced motion, the transition between turning and swimming smoother. Note that, in the swimming regime, the RL policy produces cyclic shape deformations that do not maximize the area integral of $\text{curl}_2\mathbf{A}_1$ (shape trajectory does not enclose the whole orange portion in the $\text{curl}_2\mathbf{A}_1$ image). Indeed, maximizing this integral is not optimal for forward swimming as discussed next. The key lies in the fact that the physics of the problem, specifically, the rotational motion of the fish, couples displacements in the x- and y- direction. Our model-free RL policy captures this effect with no explicit knowledge of the physics of the system.

To better explain this optimality of the RL policy, we manually prescribed cyclic shape changes that follow rectangular trajectories reminiscent of the trajectory generated by the RL policy for forward swimming. The manually-designed shape trajectories all share the same width as the RL policy albeit at different lengths, namely, 1.2 (small), 2.4 (medium), and 3.6 (large) to enclose increasingly larger regions of the orange portion of the $\text{curl}_2\mathbf{A}_1$; see right column of Fig. 5. These cyclic deformations result in net displacements in the x-direction with zero-mean excursions in the y-

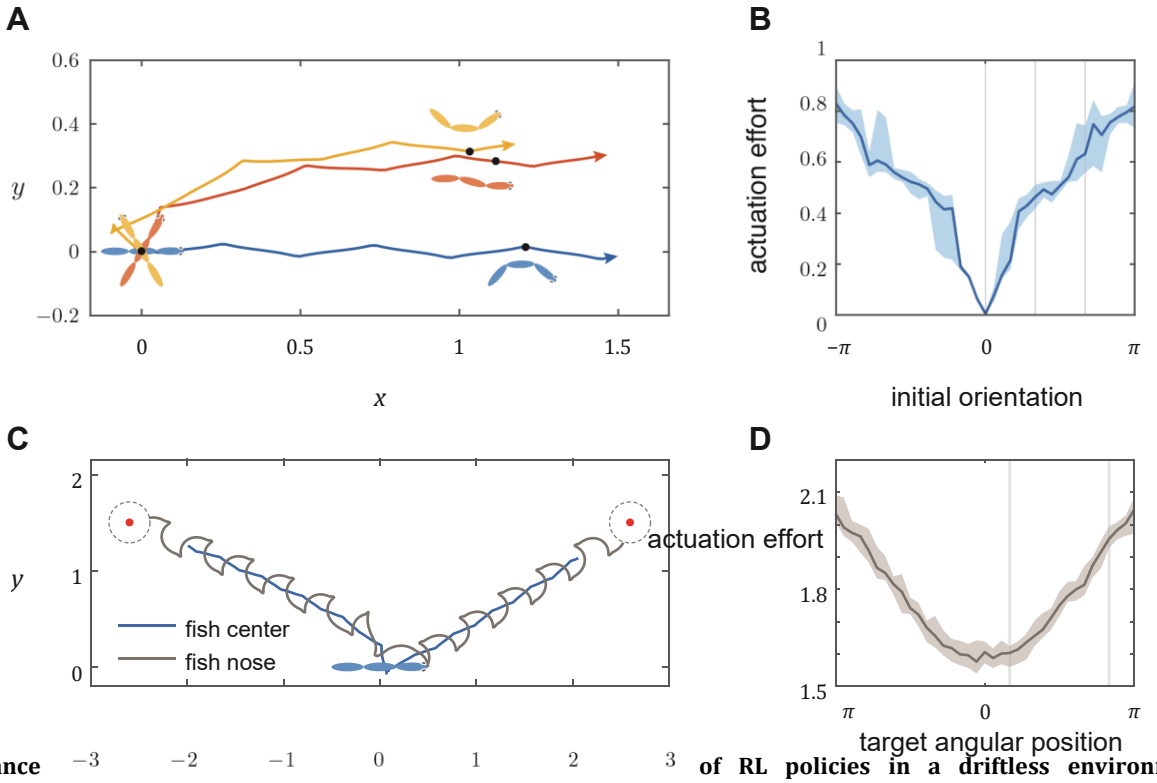

Figure 6. **Performance of RL policies in a driftless environment.** A. Swimming trajectories (center of mass) arrived at by the mean action of the direction-control policy are shown for fish starting at orientations $\beta = 0$ (blue), $\pi/3$ (red), $2\pi/3$ (yellow). B. The actuation effort of reorientation is roughly proportional to the absolute value of the initial fish orientation due to the amount of turning maneuvers required. Results shown are based on 25 stochastic policy roll-outs per tested orientation angle. C. Center of mass (blue) and nose (grey) swimming trajectories using the mean action of the naive target-seeking policy are shown with two targets located at an angular position of $\pi/6$ and $5\pi/6$. The fish is considered to have reached the target when its nose is within $\rho = 0.2$ distance from the target (dotted circles). D. Actuation effort needed to perform shape increases as the target angular position changes away from 0. Results shown use 25 stochastic policy roll-outs per tested target angle. Note that solid lines and shaded regions of B and D show the median results and variations between 25 to 75 percentile, respectively. In B, performance is calculated based on the actuation effort it takes for the fish to turn in the x-direction, while in D it is based on the effort it takes to reach the target.

direction. The y-excursions are due to the fact that, even though the area integrals of $\mathrm{curl}_2\mathbf{A}_2$ and $\mathrm{curl}_2\mathbf{A}_\beta$ over the regions enclosed by these shape trajectories are identically zero, leading to zero net rotation over a full cycle of shape deformations, the instantaneous rotations β of the fish body couple displacements in the x- and y-directions, as evident from (9). For the cyclic shape changes in Fig. 5(A), the amplitude of y-excursions is small but so is the net displacement in the x-direction; meanwhile in Fig. 5(C), both the net displacement in the x-direction and the amplitude of the y-excursions are large. The fastest fish is the one that maximizes forward motion while minimizing lateral movements, as shown in Fig. 5(B) and recapitulated in the RL result shown in Fig. 5(D). It is worth emphasizing that the RL policy arrives at this optimal solution merely by sampling observations, actions, and rewards, with no prior or developed knowledge of the physics of the problem.

Next, we investigated the effect of the initial orientation $\beta(0) \in [-\pi,\pi]$ on the amount of control effort required to turn and swim parallel to the x-axis. Fig. 6(A) shows three examples of fish following the trained policy starting from three distinct initial orientations $\beta(0) = 0, \pi/3$, and $2\pi/3$ and a straight configuration centered at the origin of the physical space. To measure the actuation effort needed in these motions, we used the integral $\int_0^\tau T_{\text{shape}} dt$ of the actuation energy $T_{\text{shape}} = \frac{1}{2}(J + m_2 a^2)(\dot{\alpha}_1^2 + \dot{\alpha}_2^2)$, which is the energy imparted to the fluid by the fish shape changes; see Appendix A. Fig. 6(B) shows the actuation effort versus the initial orientation of a straight fish. Here the fish was instructed by the stochastic policy with the same action noise as in the training process. The actuation effort, as well as its variation due to noise, is larger for larger turning angles.

Lastly, we examined the behavior and effort of a fish swimming instructed to reach a known target in an environment

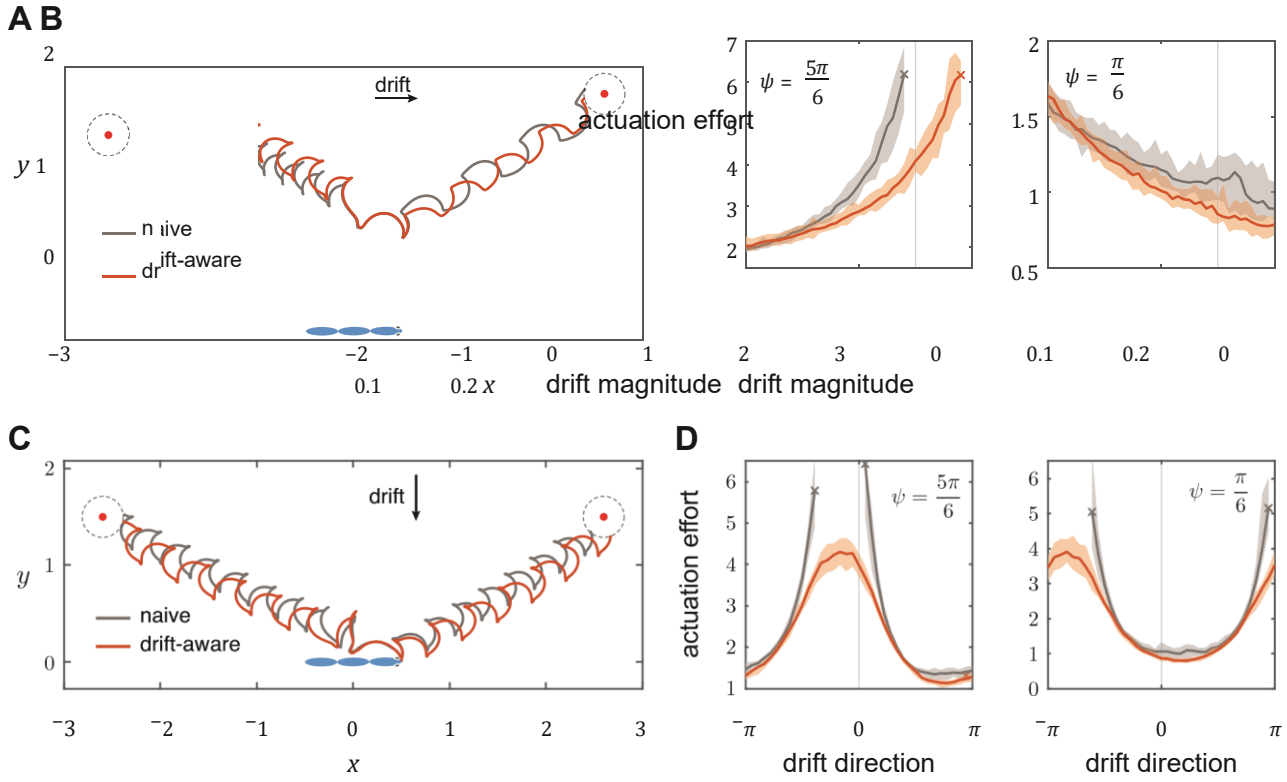

Figure 7. **Performance of target-seeking policies in the presence of drift.** We compare the swimming trajectories and actuation efforts of target-seeking policies for different drift magnitudes and direction. A and C. Naive policy (grey) and drift-aware policy (orange) produce similar average actions in environments with constant (0.1) drift in the positive x-direction and negative y-direction, respectively. Both panels showcase trajectories reaching targets located three unit-lengths away with angular position ψ of $\pi/6$ and $5\pi/6$. B. For a fixed drift direction (positive x-direction), actuation effort as a function of drift magnitude evolves differently depending on target angular position ψ. When drift is against the direction of the target (left), both policies fail to reach the target on average for large enough drift (cross marks). No failures are observed when drift facilitates swimming towards the target (right). In both cases, the drift-aware policy significantly outperforms the naive policy at large drift magnitudes by saving actuation effort. Intriguingly, inclusion of extra observations in drift-aware policies seems to result in slightly suboptimal performance when the drift magnitude is very small. D. With the drift magnitude fixed to 0.15, the naive policy fails to reach the target if the drift direction is near the opposite end of the target angular position ψ. However, at this drift magnitude, the drift-aware policy can still reach both targets regardless of

the drift direction. Note that solid lines and shaded regions of B and D show the median results and variations between 25 to 75 percentile, respectively.

with zero drift. Fig. 6(C) shows examples of the fish swimming motion in the physical space for targets located at $\psi = \pi/6$ and $\psi = 5\pi/6$. All tests ran until the fish reaches the target or a maximum interval of 1000 time steps is exceeded. We varied the target angular position while maintaining the fish initial shape and orientation (the fish always started straight in the x-direction), and we calculated the actuation effort as a function of the target orientation; see Fig. 6(D). The actuation effort increases as the target moves from the front to the back of the fish, because it requires larger turns in order for the fish to align its heading direction with the direction of the target; this is consistent with our findings based on the direction-control policy. It is worth emphasizing that the direction-control task is equivalent to the target-seeking task with the target placed at $x = +\infty$. it is also worth noting that the reward function is measured from the nose of the fish, thus breaking the fore-aft symmetry of the fish, and leading to control action that favor turning and swimming head first towards the target.

We tested the behavior and effort of the target-seeking fish in the presence of non-zero drift. Fig. 7(A) and (C) show a comparison between the naive and drift-aware policies for targets located at $\psi = \pi/6$ and $\psi = 5\pi/6$, with drift of magnitude of 0.1 pointing to the x-direction and the $-y$-direction, respectively. The naive policy (grey lines) is able to reach the target, even though it does not directly observe the drift, albeit following different actions and swimming trajectories than the drift-aware policy (orange lines). Specifically, when following the drift-aware policy, the fish curls less when the drift is helpful and curls more when the drift is unfavorable. We assessed the performance of the two policies for various drift magnitudes and directions. In Fig. 7(B), we calculated the actuation effort as a function of the drift magnitude with a fixed drift direction for two target locations. The naive policy outperforms the drift-aware policy for small drift (drift magnitudes less than 0.025) even when the drift is for adversarial, but the naive policy loses or even fails to finish the task when the drift is large, especially when the drift is in the adverse direction to the target location. This implies that it might be wise to discard some sensory input (observations) when the perturbation due to drift is weak, especially if these extra observations act more like a distraction than a clue. But as the perturbation gets stronger, it is necessary to take more observations into account. Both the naive and drift-aware policies are not able to complete the task in the given amount of time when the drift magnitude is very large and its direction is adversarial to the target location. This is because the shape actuation has no direct control over the drift itself. In Fig. 7(D), we fixed the magnitude of drift to 0.15 and changed its direction. Using the actuation effort as our performance metric as before, the drift-aware policy has better or similar performance under all tested conditions. The naive policy fails when the drift acts in the adverse direction relative to the target while the drift-aware policy is always able to reach the target before the episode terminates.

VII. CONCLUDING REMARKS

We considered a three-link fish swimming in potential flow. We reviewed that swimming in potential flow can be expressed as a combination of a dynamic phase (drift) and geometric phase (driftless) over the shape of fish body deformation [44]. In the driftless case, net locomotion is purely determined by the fish shape deformations, and geometric techniques can be used for gait design and motion planning over the fish shape space [52, 54], but shape actuation cannot control the drift itself. Yet, even in the driftless regime, motion planning starting from arbitrary fish orientation and shape is not trivial. In this paper, we applied model-free reinforcement learning techniques for controlling the fish motion, and we arrived at optimal policies for swimming (i) in a desired direction, and (ii) towards a target in the absence and presence of drift. The RL based policies produce behavior that is robust to variations in the fish initial shape and orientation and target location. We used the actuation effort as a measure of the policy performance under various initial conditions and in various environments, and we quantified the robustness of the RL policies to the presence of drift. We found that although the fish has no control over the drift itself, the fish learns to take advantage of the presence of moderate drift to reach its target. Therefore, it might be wise to discard some sensory input (observations) for weak signals because these extra observations could act more like a distraction than a clue. We also found that for small drift, the drift-naive policy outperforms the drift-aware policy even when drift is adversarial to the target location. However, large adversarial drift hinders the fish ability to locate the target. Importantly, these insights into the RL policies were achievable by combining tools from geometric mechanics with RL-based control. Geometric tools such as the concept of shape space, combined with the RL notion of the action space, provide a useful and novel framework for visualizing and interpreting the RL policies as action vector fields over the shape space.

A few comments on the advantages and limitations of our implementation are in order. Despite algorithmic advantages, obtaining an RL policy is computationally costly, especially when the environment simulator involves high-fidelity fluid-structure interaction models. To circumvent this problem, recent work on training fish to swim uses a limited set of observations and actions [39]. For example, the zebrafish model of [39] allows only 5 discrete actions, each corresponding to a fixed amplitude of body curvature change. Reduced order fluid models, such as the potential flow model used here, offer an enticing framework for designing control laws that can later be tested and refined using more realistic flow environments, as done in [20] for a manually-designed swimming gait in [44]. Specifically, in the simplified potential flow environment employed here, we are able to train continuous action policies using a rich set of observations, with an eye on probing the performance of these policies in more realistic flow environments in future work.

ACKNOWLEDGMENTS

Kanso would like to acknowledge support from the Office of Naval Research through the grants N00014-17-1-2287 and N00014-17-1-2062. The authors are thankful for useful discussions with Dr. Yuval Tassa.

[1] Dickinson MH, Lehmann FO, Sane SP. Wing rotation and the aerodynamic basis of insect flight. Science. 1999;284(5422):1954–1960.
[2] Tytell E, Holmes P, Cohen A. Spikes alone do not behavior make: why neuroscience needs biomechanics. Current opinion in neurobiology. 2011;21(5):816–822.
[3] Merel J, Botvinick M, Wayne G. Hierarchical motor control in mammals and machines. Nature Communications. 2019;10(1):1–12.
[4] Madhav MS, Cowan NJ. The synergy between neuroscience and control theory: the nervous system as inspiration for hard control challenges. Annual Review of Control, Robotics, and Autonomous Systems. 2020;3:243–267.
[5] Huang KH, Ahrens MB, Dunn TW, Engert F. Spinal projection neurons control turning behaviors in zebrafish. Current Biology. 2013;23(16):1566–1573.
[6] Burgess HA, Granato M. Sensorimotor gating in larval zebrafish. Journal of Neuroscience. 2007;27(18):4984–4994.
[7] Privat M, Romano SA, Pietri T, Jouary A, Boulanger-Weill J, Elbaz N, et al. Sensorimotor Transformations in the Zebrafish Auditory System. Current Biology. 2019;29(23):4010–4023.
[8] Marchese AD, Onal CD, Rus D. Autonomous Soft Robotic Fish Capable of Escape Maneuvers Using Fluidic Elastomer Actuators. Soft Robotics. 2014;1(1):75–87. PMID: 27625912. Available from: https://doi.org/10.1089/soro.2013.0009.
[9] Lauder GV. Fish Locomotion: Recent Advances and New Directions. Annual Review of Marine Science. 2015;7(1):521–545. PMID: 25251278. Available from: https://doi.org/10.1146/annurev-marine-010814-015614.
[10] Cianchetti M, Follador M, Mazzolai B, Dario P, Laschi C. Design and development of a soft robotic octopus arm exploiting embodied intelligence. In: 2012 IEEE International Conference on Robotics and Automation. IEEE; 2012. p. 5271–5276.
[11] Laschi C, Cianchetti M, Mazzolai B, Margheri L, Follador M, Dario P. Soft robot arm inspired by the octopus. Advanced Robotics. 2012;26(7):709–727.
[12] Pfeifer R, Bongard J. How the body shapes the way we think: a new view of intelligence. MIT press; 2006.
[13] Pfeifer R, Lungarella M, Iida F. Self-Organization, Embodiment, and Biologically Inspired Robotics. Science. 2007;318(5853):1088–1093. Available from: https://science.sciencemag.org/content/318/5853/1088.
[14] Van Rees WM, Gazzola M, Koumoutsakos P. Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers. Journal of Fluid Mechanics. 2013;722.
[15] Liao JC, Beal DN, Lauder GV, Triantafyllou MS. Fish exploiting vortices decrease muscle activity. Science. 2003;302(5650):1566–1569.
[16] Mu¨ller UK. Fish'n flag. Science. 2003;302(5650):1511–1512.
[17] Eloy C. On the best design for undulatory swimming. Journal of Fluid Mechanics. 2013;717:48–89.
[18] Kern S, Koumoutsakos P. Simulations of optimized anguilliform swimming. Journal of Experimental Biology. 2006;209(24):4841–4857.
[19] Mittal R, Dong H, Bozkurttas M, Loebbecke A, Najjar F. Analysis of flying and swimming in nature using an immersed boundary method. In: 36th AIAA Fluid Dynamics Conference and Exhibit; 2006. p. 2867.
[20] Eldredge JD. Numerical simulations of undulatory swimming at moderate Reynolds number. Bioinspiration & biomimetics. 2006;1(4):S19.
[21] Gazzola M, Argentina M, Mahadevan L. Gait and speed selection in slender inertial swimmers. Proceedings of the National Academy of Sciences. 2015;112(13):3874–3879.

[22] Tytell ED, Hsu CY, Williams TL, Cohen AH, Fauci LJ. Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming. Proceedings of the National Academy of Sciences. 2010;107(46):19832–19837.
[23] Hamlet C, Fauci LJ, Tytell ED. The effect of intrinsic muscular nonlinearities on the energetics of locomotion in a computational model of an anguilliform swimmer. Journal of theoretical biology. 2015;385:119–129.
[24] Guthrie D. Role of vision in fish behaviour. In: The behaviour of Teleost fishes. Springer; 1986. p. 75–113.
[25] Fernald RD. Fish vision. In: Development of the vertebrate retina. Springer; 1989. p. 247–265.
[26] Douglas R, Djamgoz M. The visual system of fish. Springer Science & Business Media; 2012.
[27] Engelmann J, Hanke W, Mogdans J, Bleckmann H. Hydrodynamic stimuli and the fish lateral line. Nature. 2000;408(6808):51–52. Available from: https://doi.org/10.1038/35040706.
[28] Ristroph L, Liao JC, Zhang J. Lateral Line Layout Correlates with the Differential Hydrodynamic Pressure on Swimming Fish. Phys Rev Lett. 2015 Jan;114:018102. Availablefrom: https://link.aps.org/doi/10.1103/PhysRevLett.114.018102.
[29] Colvert B, Kanso E. Fishlike rheotaxis. Journal of Fluid Mechanics. 2016;793:656–666.
[30] Colvert B, Liu G, Dong H, Kanso E. How can a source be located by responding to local information in its hydrodynamic trail? In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE; 2017. p. 2756–2761.
[31] Montgomery JC, Baker CF, Carton AG. The lateral line can mediate rheotaxis in fish. Nature. 1997;389(6654):960–963. Available from: https://doi.org/10.1038/40135.
[32] Filella A, Nadal F, Sire C, Kanso E, Eloy C. Model of collective fish behavior with hydrodynamic interactions. Physical review letters. 2018;120(19):198101.
[33] Li L, Nagy M, Graving JM, Bak-Coleman J, Xie G, Couzin ID. Vortex phase matching as a strategy for schooling in robots and in fish. Nature communications. 2020;11(1):1–9.
[34] Partridge BL, Pitcher TJ. The sensory basis of fish schools: relative roles of lateral line and vision. Journal of comparative physiology. 1980;135(4):315–325. Available from: https://doi.org/10.1007/BF00657647.
[35] Katz Y, Tunstrøm K, Ioannou CC, Huepe C, Couzin ID. Inferring the structure and dynamics of interactions in schooling fish. Proceedings of the National Academy of Sciences. 2011;108(46):18720–18725.
[36] Gazzola M, Tchieu AA, Alexeev D, de Brauer A, Koumoutsakos P. Learning to school in the presence of hydrodynamic interactions. Journal of Fluid Mechanics. 2016;789:726–749.
[37] Gustavsson K, Biferale L, Celani A, Colabrese S. Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning. The European Physical Journal E. 2017;40(12):110.
[38] Colabrese S, Gustavsson K, Celani A, Biferale L. Flow navigation by smart microswimmers via reinforcement learning. Physical review letters. 2017;118(15):158004.
[39] Verma S, Novati G, Koumoutsakos P. Efficient collective swimming by harnessing vortices through deep reinforcement learning. Proceedings of the National Academy of Sciences. 2018;115(23):5849–5854.
[40] Colvert B, Alsalman M, Kanso E. Classifying vortex wakes using neural networks. Bioinspiration & biomimetics. 2018;13(2):025003.
[41] Alsalman M, Colvert B, Kanso E. Training bioinspired sensors to classify flows. Bioinspiration & Biomimetics. 2018 nov;14(1):016009. Available from: https://doi.org/10.1088%2F1748-3190%2Faaef1d.
[42] Weber P, Arampatzis G, Novati G, Verma S, Papadimitriou C, Koumoutsakos P. Optimal Flow Sensing for Schooling Swimmers. Biomimetics. 2020;5(1):10.
[43] Gazzola M, Hejazialhosseini B, Koumoutsakos P. Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. SIAM Journal on Scientific Computing. 2014;36(3):B622–B639.
[44] Kanso E, Marsden JE. Optimal motion of an articulated body in a perfect fluid. In: Proceedings of the 44th IEEE Conference on Decision and Control. IEEE; 2005. p. 2511–2516.
[45] Kanso E, Marsden JE, Rowley CW, Melli-Huber JB. Locomotion of articulated bodies in a perfect fluid. Journal of Nonlinear Science. 2005;15(4):255–289.
[46] Dickinson MH, Farley CT, Full RJ, Koehl M, Kram R, Lehman S. How animals move: an integrative view. science. 2000;288(5463):100–106.
[47] Degris T, Pilarski PM, Sutton RS. Model-Free reinforcement learning with continuous action in practice. In: 2012 American Control Conference (ACC); 2012. p. 2177–2182.
[48] Haith AM, Krakauer JW. Model-Based and Model-Free Mechanisms of Human Motor Learning. In: Richardson MJ, Riley MA, Shockley K, editors. Progress in Motor Control. New York, NY: Springer New York; 2013. p. 1–21.
[49] Tsang ACH, Tong PW, Nallan S, Pak OS. Self-learning how to swim at low Reynolds number. Physical Review Fluids. 2020;5(7):074101.
[50] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.
[51] Eldredge JD. Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method. Journal of Computational Physics. 2007;221(2):626 – 648. Available from: http://www.sciencedirect.com/science/article/pii/S0021999106003093.
[52] Melli JB, Rowley CW, Rufat DS. Motion planning for an articulated body in a perfect planar fluid. SIAM Journal on applied dynamical systems. 2006;5(4):650–669.
[53] Nair S, Kanso E. Hydrodynamically coupled rigid bodies. Journal of Fluid Mechanics. 2007;592:393–411.

[54] Hatton RL, Choset H. Connection vector fields and optimized coordinates for swimming systems at low and high Reynolds numbers. In: ASME 2010 Dynamic Systems and Control Conference. American Society of Mechanical Engineers Digital Collection; 2010. p. 817–824.
[55] Morgansen KA, Duidam V, Mason RJ, Burdick JW, Murray RM. Nonlinear control methods for planar carangiform robot fish locomotion. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164). vol. 1. IEEE; 2001. p. 427–434.
[56] Bloch AM, Krishnaprasad P, Marsden JE, Murray RM. Nonholonomic mechanical systems with symmetry. Archive for Rational Mechanics and Analysis. 1996;136(1):21–99.
[57] Murray RM, Li Z, Sastry SS, Sastry SS. A mathematical introduction to robotic manipulation. CRC press; 1994.
[58] Kelly SD, Pujari P, Xiong H. Geometric mechanics, dynamics, and control of fishlike swimming in a planar ideal fluid. In: Natural Locomotion in Fluids and on Surfaces. Springer; 2012. p. 101–116.
[59] Lamb H. Hydrodynamics. Dover Books on Physics. Dover publications; 1945. Available from: https://books.google.com/books?id=237xDg7T0RkC.
[60] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning. 1992;8(3-4):229–256.
[61] Sutton RS, McAllester DA, Singh SP, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems; 2000. p. 1057–1063.
[62] Kakade SM. A natural policy gradient. In: Advances in neural information processing systems; 2002. p. 1531–1538.
[63] Peters J, Schaal S. Natural actor-critic. Neurocomputing. 2008;71(7-9):1180–1190.
[64] Peters J, Schaal S. Policy gradient methods for robotics. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE; 2006. p. 2219–2225.
[65] Konda VR, Tsitsiklis JN. Actor-critic algorithms. In: Advances in neural information processing systems; 2000. p. 1008–1014.
[66] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:180101290. 2018;.
[67] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv preprint arXiv:170706347. 2017;.
[68] Leonard NE. Stability of a bottom-heavy underwater vehicle. Automatica. 1997;33(3):331–346.
[69] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

Appendix A: Physics of the fish model

We review the derivation of the equations of motion governing the swimming of an articulated three-link fish in potential flow (Fig. 1).

1. Fish kinematics

Consider planar motions of the three-link fish. Let $\mathbf{x} = (x,y)$ denote the position of the center of mass G of the middle link, and let β denote the orientation of the fish relative to a fixed inertial frame, here taken to be the angle between the x-axis and the major axis of symmetry of the middle link. Let α_1 and α_2 be the rotation angles of the front link relative to the middle link and the middle link relative to the rear link, that is to say, (α_1,α_2) represents the shape of the three-link fish. It is convenient for the following development to introduce a body-fixed frame $(\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3)$, attached at G and co-rotating with the middle link. This body-fixed frame is related to the inertial frame $(\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3)$ via a rigid-body rotation such that $\mathbf{e}_1 = \cos\beta\mathbf{b}_1 - \sin\beta\mathbf{b}_2$, $\mathbf{e}_2 = \sin\beta\mathbf{b}_1 + \cos\beta\mathbf{b}_2$, and $\mathbf{e}_3 = \mathbf{b}_3$.

The velocity $(\dot{x},\dot{y})$ of the center of mass of the middle link, when expressed in the body-fixed frame, is given by

$$\mathbf{v} = v_1\mathbf{b}_1 + v_2\mathbf{b}_2 = (\dot{x}\cos\beta + \dot{y}\sin\beta)\mathbf{b}_1 + (-\dot{x}\sin\beta + \dot{y}\cos\beta)\mathbf{b}_2 \qquad \text{(A1)}$$

Assuming all three links are made of identical ellipsoids of half-length a, half-width b, and half-height c, the velocities of the centers of mass G_1 and G_2 of the front and rear link, expressed in the body-fixed frame of the middle link, are given by (i = 1,2, denote the front and rear link, respectively)

$$\mathbf{v}_i = (v_1 \mp a\dot{\alpha}_i\sin\alpha_i - a^{\beta\cdot}\sin\alpha_i)\mathbf{b}_1 + (v_2 \pm a^{\beta\cdot} + a\dot{\alpha}_i\cos\alpha_i \pm a^{\beta\cdot}\cos\alpha_i)\mathbf{b}_2. \qquad \text{(A2)}$$

The angular velocities of the middle, front, and rear links are given by $\dot{\beta}$, $\dot{\beta} + \dot{\alpha}_1$, and $\dot{\beta} - \dot{\alpha}_2$ respectively. For our simulations, we used $a : b : c = 5 : 1 : 5$ as the geometry of the ellipsoids.

2. Kinetic energy of the articulated body

In the absence of the fluid, the kinetic energy of the articulated three-link body is given by

$$T_{\text{body}} = \frac{1}{2} m_s \mathbf{v} \cdot \mathbf{v} + \frac{1}{2} J_s \dot{\beta}^2 + \frac{1}{2} \sum_i \left[m_s \mathbf{v}_i \cdot \mathbf{v}_i + J_s (\dot{\beta} \pm \dot{\alpha}_i)^2 \right] \tag{A3}$$

where $m_s = \frac{4}{3}\pi abc\rho_s$ and $J_s = \frac{1}{5}(a^2 + b^2)m_s$ are the mass and moment of inertia of each solid link with ρ_s the density of the links.

3. Kinetic energy of the fluid

The three-link fish is submerged in an unbounded domain of incompressible and irrotational fluid, such that the fluid velocity **u** = ∇*φ* can be expressed as the gradient of a potential function *φ*. It is a standard result in potential flow theory that the kinetic energy of the fluid can be expressed in terms of the variables of the submerged solid [44, 45, 59].

In the case of a single ellipsoid, the kinetic energy of the fluid is given by $T_{\text{fluid}} = [(m_{1a} v_1^2 + m_{2a} v_2^2) + J_a \dot{\beta}^2]/2$, where m_{1a}, m_{2a} and J_a are the added mass and added moment of inertia due to the presence of the fluid, expressed in a body-fixed frame that coincides with the major and minor axes of the ellipsoid. These quantities depend on the geometric properties *a,b,c* of the submerged ellipsoid, as are given in the Appendix B of [68]. For a non-spherical body, the added masses m_{1a}, m_{2a} depend on the direction of motion: the added mass is larger when moving in the direction of the minor axis of symmetry of the ellipsoid, that is to say, in the transverse direction, hence $m_{1a} \le m_{2a}$. In the case of the three-link fish, the kinetic energy of the fluid is of the form

$$\begin{aligned} T_{\text{fluid}} = & \frac{1}{2} m_{1a} v_1^2 + \frac{1}{2} m_{2a} v_2^2 + \frac{1}{2} J_a \dot{\beta}^2 + \frac{1}{2} \sum_i J_a \left(\dot{\beta} \pm \dot{\alpha}_i \right)^2 \\ & + \frac{1}{2} \sum_i m_{1a} \left(v_1 \cos\alpha_i \pm v_2 \sin\alpha_i + a\dot{\beta} \sin\alpha_i \right)^2 \\ & + \frac{1}{2} \sum_i m_{2a} \left(\mp v_1 \sin\alpha_i + v_2 \cos\alpha_i \pm a\dot{\beta} \cos\alpha_i \pm a\dot{\beta} + a\dot{\alpha}_i \right)^2 . \end{aligned} \tag{A4}$$

Here we transform the velocity components of head and tail by α_1 and α_2, respectively, to match the added mass components.

4. Kinetic energy of the body-fluid system

The kinetic energy of the fish-fluid system is obtained by taking the sum of Eq. A3 and Eq. A4, which can be expressed in matrix form as follows:

$$T = T_{\text{body}} + T_{\text{fluid}} = \frac{1}{2} \begin{bmatrix} v_1 & v_2 & \dot{\beta} \end{bmatrix} \begin{bmatrix} I_{\text{lock}} & I_{\text{couple}} \\ I_{\text{couple}}^{\mathsf{T}} & I_{\text{shape}} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \dot{\beta} \end{bmatrix}, \tag{A5}$$

$$\begin{bmatrix} \dot{\alpha}_1 & \dot{\alpha}_2 \end{bmatrix}$$

Here, I_{lock} is a 3 × 3 locked mass matrix, function of α_1 and α_2,

$$I_{\text{lock}} = \begin{bmatrix} \mathbb{M} & \mathbb{H} \\ \mathbb{H}^{\mathsf{T}} & \mathbb{J} \end{bmatrix}, \tag{A6}$$

where M is a 2 × 2 mass matrix given by

$$\mathbb{M} = \begin{bmatrix} m_1\left(1+\sum_i \cos^2\alpha_i\right) + m_2\sum_i \sin^2\alpha_i & \frac{1}{2}(m_1-m_2)(\sin 2\alpha_1 - \sin 2\alpha_2) \\ \frac{1}{2}(m_1-m_2)(\sin 2\alpha_1 - \sin 2\alpha_2) & m_2\left(1+\sum_i \cos^2\alpha_i\right) + m_1\sum_i \sin^2\alpha_i \end{bmatrix}, \tag{A7}$$

J is a moment-of-inertia scalar given by

$$\mathbb{J} = 3J + m_1 a^2 \sum_i \sin^2\alpha_i + m_2 a^2 \sum_i (1+\cos\alpha_i)^2 \tag{A8}$$

and H is given by

$$\mathbb{H} = \begin{bmatrix} \frac{1}{2}(m_1-m_2)a\sum_i \sin 2\alpha_i - m_2 a\sum_i \sin\alpha_i \\ \frac{1}{2}(m_1-m_2)a(\cos 2\alpha_2 - \cos 2\alpha_1) + m_2 a(\cos\alpha_1 - \cos\alpha_2) \end{bmatrix}. \tag{A9}$$

Here we used $m_1 = m_s + m_{1a}$, $m_2 = m_s + m_{2a}$, and $J = J_s + J_a$. Note that H couples the translational and rotational motion of the articulated body. In the case of single ellipsoid, H is identically zero.

Further, $\mathbb{I}_{\text{couple}}$ is a 3 × 2 matrix that couples rigid body motion with shape deformation,

$$\mathbb{I}_{\text{couple}} = \begin{bmatrix} -m_2 a\sin\alpha_1 & m_2 a\sin\alpha_2 \\ m_2 a\cos\alpha_1 & m_2 a\cos\alpha_2 \\ J + m_2 a^2(1+\cos\alpha_1) & -J - m_2 a^2(1+\cos\alpha_2) \end{bmatrix}. \tag{A10}$$

Finally, $\mathbb{I}_{\text{shape}}$ is a 2 × 2 matrix associated with shape deformation,

$$\mathbb{I}_{\text{shape}} = \begin{bmatrix} J + m_2 a^2 & 0 \\ 0 & J + m_2 a^2 \end{bmatrix}. \tag{A11}$$

The total linear and angular momenta P_1, P_2, and Π expressed in body frame are given by $P_1 = \partial T/\partial v_1$, $P_2 = \partial T/\partial v_2$, $\Pi = \partial T/\partial \dot{\beta}$ to arrive at (3) of the main text

$$\begin{bmatrix} P_1 \\ P_2 \\ \Pi \end{bmatrix} = \mathbb{I}_{\text{lock}} \begin{bmatrix} v_1 \\ v_2 \\ \dot{\beta} \end{bmatrix} + \mathbb{I}_{\text{couple}} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix}. \tag{A12}$$

Algorithm B.1 Environment Simulation

1: **for** time step $t = 0,1,\ldots$ **do**
2: **if** $t = 0$ **or** episode terminates **then**
3: store time step of episode termination,
4: reset state $s_t \sim P(s_0)$
5: evaluate observation: $o_t \sim o(s_t)$
6: **end if**
7: sample action from policy $a_t \sim \pi_\theta(a_t|o_t)$
8: evolve next state according to fish physics $s_{t+1} \sim P(s_{t+1}|s_t,a_t)$
9: evaluate next observation $o_{t+1} \sim o(s_{t+1})$ and reward $r_t \sim r(a_t,o_{t+1})$
10: **if** $t = 0$ **or** mod(t,N) 6= 0 **then**
11: append current action, observation, reward, and probability of sampling the action to assemble vectors $a_{N\times n_a}$, $o_{N\times n_o}$, $r_{N\times 1}$, and $\pi_{\theta_{\text{old}}}(a|o)_{N\times 1}$
12: **else**
13: update agent networks according to Algorithm B.2
14: **end if**
15: **end for**

Algorithm B.2 Updating the Agent

for update epoch number $\kappa = 0,1,\ldots K$ **do**
  compute the truncated return using rewards $r_{N\times 1}$ and assemble into vector $R_{N\times 1}$
  estimate infinite-horizon return using $R_{N\times 1}$ and $V_T = V_\varphi(o_T)$ if bootstrapping is desired (see Eq.B2)
  using $o_{N\times n_o}$ and value function V_φ, evaluate expected returns at each time step and store into $V_{N\times 1}$
  compute the advantage $A = R_{N\times 1} - V_{N\times 1}$ and normalize by its mean and variance if desired
  evaluate the probability of realizing $a_{N\times n_a}$ based on $o_{N\times n_o}$ for the policy π_θ, and store to $\pi_\theta(a|o)_{N\times 1}$
  compute the action-likelihood ratio: $\varrho_\theta = \dfrac{\pi_\theta(a|o)_{N\times 1}}{\pi_{\theta_{\text{old}}}(a|o)_{N\times 1}}$
  compute clipped surrogate loss function: $L_{\text{clip}}(\theta) = \text{mean}[\min[_\theta \cdot A, \text{clip}(_\theta, 1-\varrho, 1+\varrho)\cdot A]]$
  compute the value-function loss: $L_{\text{value}}(\varphi) = 0.5 \cdot \text{mean}\left[(R_{N\times 1} - V_{N\times 1})^2\right]$
  compute the total loss: $L(\theta,\varphi) = -L_{\text{clip}}(\theta) + L_{\text{value}}(\varphi) - \alpha \cdot \text{entropy}[\pi_\theta]$
update parameters (θ,φ) to minimize the total loss using a gradient based optimizer (*e.g.*, Adam [69]) **end for**

Appendix B: Proximal Policy Optimization (PPO) Algorithms

We implement the clipped advantage Proximal Policy Optimization (PPO) method proposed by [67] for our RL training. PPO maximizes a surrogate objective that clips off unwanted changes when the policy deviates too much from the policy of the previous cycle to ensure faster and more robust convergence. We refer readers to the original reference cited above as well as the OpenAI's documentation of the PPO algorithm and their baseline implementations for a thorough explanation of the theory and details behind this method.

Our implementation can be separated into two parts. The main loop simulates the environment using action sequence a_t generated by the agent, and stores the observed rollouts for future updates; see Algorithm B.1. Note that n_o and n_a are used to indicate the number of observable states and actions. Equations describing the fish-fluid interactions were integrated numerically using adaptive time step, explicit RK45 method between each decision step of 0.1 unit of time. This choice of decision time step-size limits the maximum rotation allowed for the fish head and tail to be 0.1 radian per step.

Parameters of the actor-critic networks of the RL agent are updated every N time steps for K epochs. Here the value of K is chosen to be 80 and the value of N is set to 4050, an integer multiple of the episode length 150. For simplicity, we assume our continuous action variables follows a multivariate normally-distributed policy π_θ with mean value represented by a neural network parameterized by θ and constant diagonal covariance matrices, and the critic / value function $V_\varphi(o_t)$ is also represented by a neural network with parameters φ. Specifically, both the mean policy and value function are implemented as feed-forward neural network with two hidden layers and tanh activation functions. The sizes of the two hidden layers were fixed to 64 and 32, respectively. Each diagonal entry of the covariance matrix is set to 0.5^2. Finally, using the collected trajectories during the previous N time steps, the parameters θ,φ are

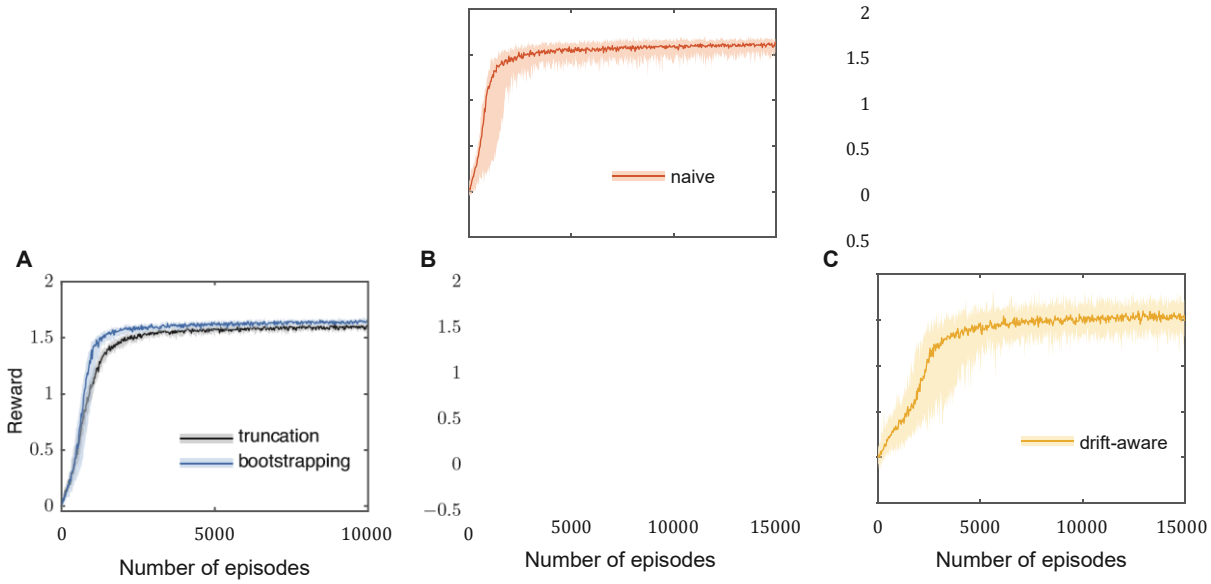

Figure B.1. **Evolution of rewards during the training process.** A. Total rewards per episode achieved by policies trained to swim parallel to the *x*-axis in a driftless environment using bootstrapped (blue) and truncated (black) return estimates. Here solid lines indicate the median, and the shaded region shows the variation between 25-75 percentile for 24 runs of the learning algorithm. B. and C. Total rewards obtained by policies trained to swim towards a given target, both of which adopt bootstrapped return estimates. Red in B reresents naive policies trained in driftless environment, while yellow in C represents policies trained in the presence of drift, with drift magnitude and direction supplied as additional observations to the policy. Again, lines and shaded regions indicate median and 25-75 percentile range respectively.

updated according a total loss function L(θ,φ) via a back-propagating gradient based optimizer; see Algorithm B.2. Note that since we did not perform systematic hyper-parameter tuning, readers might want to explore different values for better performance.

Another important side-note is that since it is in general impossible to obtain unrealized infinite horizon return $R_t = \Sigma_{t'}^{\infty} \gamma^{t'-t} r_{t'}$, we need to choose an appropriate estimator of this value based on finite length simulations. We can either simply truncate rewards after some step *k* by using

$$\hat{R}_t\Big|_{\text{truncation}} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{k-1} r_{t+k}, \tag{B1}$$

or we can use the trained value function (critic) to approximate the residual contribution to the return via *k*-step bootstrapping

$$\hat{R}_t\Big|_{\text{bootstrapping}} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^k V_\varphi(o_{t+k+1}). \tag{B2}$$

We compared these two approaches for the direction-based task and observed that bootstrapping results in faster convergence and higher rewards in general; see Fig. B.1(A). As a result, bootstrapping is used for all tasks depicted in the main text, where *k* was determined by the number of available future rewards. Namely, *k* decreases from 149 to 0 as the number of time steps increased from 1 to 150 in each episode. In addition, We show the difference in training rewards and convergence speed between the naive policy and the drift-aware policy in Fig. B.1(B). In general, the inclusion of more observations increased the time to convergence and variance in training rewards.

Lastly, we invite interested readers to visit our source repository at `https://github.com/mjysh/RL3linkFish` for the complete details of our implementation.