Deep reinforcement learning enabling a BCFbot to learn various undulatory patterns

Hameed, Imran; Chao, Xu; Navarro-Alarcon, David; Jing, Xingjian

Published in:
Ocean Engineering

Published: 15/03/2025

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Published version (DOI):
10.1016/j.oceaneng.2025.120322

Publication details:
Hameed, I., Chao, X., Navarro-Alarcon, D., & Jing, X. (2025). Deep reinforcement learning enabling a BCFbot to learn various undulatory patterns. *Ocean Engineering*, *320*, Article 120322. https://doi.org/10.1016/j.oceaneng.2025.120322

Ocean Engineering 320 (2025) 120322

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Research paper

Deep reinforcement learning enabling a BCFbot to learn various undulatory patterns

Imran Hameed [a,b], Xu Chao [a,b], David Navarro-Alarcon [a], Xingjian Jing [b,*]

[a] Department of Mechanical Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong, China
[b] Department of Mechanical Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong, China

ARTICLE INFO

Index Terms:
Biomimetic marine robots
Swimming gait optimization
Deep reinforcement learning

ABSTRACT

In bio-inspired marine robots, one particular motion pattern is generally adopted to achieve benefits of that pattern. However, multiple gait patterns can be utilized together in a single biomimetic design to employ their benefits, as required. However, there is a lack of a unified control scheme that can be used to optimize and mimic undulatory patterns observed among different organisms in the body and/or caudal fin (BCF) category. Thus, central pattern generators (CPGs) were incorporated into a deep reinforcement learning (DRL) architecture to train a robot to develop various swimming gaits. The proposed framework can not only develop and optimize distinct motion patterns but also seamlessly and instantly switch between them. Oscillators integrated into a learning paradigm provide a bioinspired framework to systematically develop a variety of swimming gaits. The prototyped BCFbot has multiple joints, which makes it easy to realize more than one tail undulation patterns. Three different swimming patterns (anguilliform, sub-carangiform, and carangiform) were learned through simulation and then verified on a physical robot. Testing and comparison results show that the claimed benefits of the three benchmark motion patterns can be well realized using the developed robot and can be freely switched and optimized using the developed DRL mechanism. This should be the first attempt for achieving a multimotion pattern optimization and switching within a single BCFbot and demonstrating a successful motion generation regime similar to a real animal.

1. Introduction

Marine robots can be broadly divided into two categories: propeller-based robots relying on screw-type thrusters (Singh et al., 2017/06) or fan-wing propellers (Gao et al., 2021/12), and bioinspired designs mimicking marine creatures (Raj and Thakur, 2016/04; Wang et al., 2022). Bioinspired marine robots have undulating and/or oscillating surfaces inspired by marine creatures. They use tentacles (Fras et al., 2018), caudal (Katzschmann et al., 2018/03), or pectoral (Meng et al., 2022) fins to produce propulsion. Compared to propeller-based robots, bio-inspired designs cause less vibrations and noise in marine environment and provide better maneuverability (Katzschmann et al., 2018/03). Due to absence of fast-moving parts, they are considered safe for the marine life. They can be made using soft materials to withstand high pressure (Li et al., 2021/03). They can be propelled and maneuvered using a single propulsor (Kopman and Porfiri, 2013).

1.1. BCF locomotion in nature

Marine biologists have classified the propulsion phenomena in marine creatures into two broad and distinctive categories: body and caudal fin (BCF) propulsion, and median and paired fin (MPF) propulsion (Lindsey, 1978). BCF swimmers exhibit different swimming modes, depending on their body shape, size, and flexibility. Although the swimming gaits of BCF swimmers appear visibly different, they are difficult to differentiate quantitatively. A recent study (Di Santo et al., 2021) (Fig. 1) showed that the amplitudes of the constituent members are either the same or slightly different, and that the wavelength is a critical factor that can be used to differentiate them.

In Fig. 1, the left-hand side shows the anguilliform mode, in which the average wavelength is the lowest. An example is *Anguilla rostrata* (American eel). Swimmers with long and slender bodies, such as snakes and eels, exhibit this motion pattern. The other categories have fusiform-shaped bodies. Depending on the flexibility of their bodies, they exhibit sub-carangiform to carangiform motion patterns. For example,

* Corresponding author.
E-mail addresses: xingjing@cityu.edu.hk, xingjian.jing@gmail.com (X. Jing).

https://doi.org/10.1016/j.oceaneng.2025.120322
Received 7 March 2024; Received in revised form 7 December 2024; Accepted 5 January 2025
Available online 11 January 2025

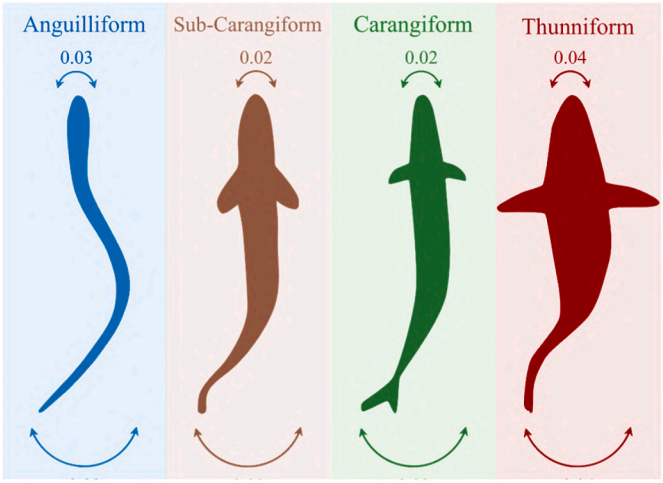

Fig. 1. Canonical swimming patterns included in the BCF category. Visual representations are originally from (Katzschmann et al., 2018/03) and amplitude information is taken from (Meng et al., 2022).

Salvalinus fontinalis (trout) exhibits a sub-carangiform pattern, and *Scomber scombrus* (mackerel) exhibits a carangiform pattern. The trend for wavelength increases from the anguilliform to sub-carangiform and carangiform swimmers.

There are certain benefits associated with all types of swimming gaits. Anguilliform swimmers, for instance, are generally slow, but some can move backward freely (e.g., eel (D'AoUT and and Aerts, 1999)), whereas carangiform swimmers cannot. Carangiform swimmers move relatively faster, and their average linear speed for forward locomotion is relatively higher than that of anguilliform swimmers; for example, scombroids can swim at speeds higher than 1 m/s (Lindsey, 1978).

1.2. BCF locomotion in robotics

On the bio-inspired side of marine robotics, certain motion patterns have been adopted for robotic platforms to exploit the benefits related to these patterns. For example, the platform in (Niu et al., 2014) can move forward and backward using the anguilliform locomotion pattern, and the robotic fish in (Clapham and Hu, 2014) can swim at high speeds using the carangiform mode.

Numerous efforts have been made to mimic the anguilliform (Crespi and Ijspeert, 2008; Niu et al., 2013/09), sub-carangiform (Salumä et al., 2013; Daou et al., 2011), and carangiform (Clapham and Hu, 2014; Farideddin et al., 2015) gaits on BCF-type robots. However, most existing studies have focused on a particular motion pattern or feature related to one type of swimmer. In this study, we developed multiple motion patterns on the same platform to capture the benefits associated with different patterns.

Travelling-wave kinematics based on Lighthill's work (Lighthill, 1960), also known as the fish body wave kinematics, have been extensively studied and applied on fish-like robots. The parameters of the aforementioned model can be manually tuned to obtain different waveforms. For example, carangiform motion patterns have been studied on robotic fishes (Farideddin et al., 2015; Barrett et al., 1996; Junzhi et al., 2004; Yan et al., 2008/06) using travelling-wave kinematics. Different patterns can also be established by capturing the kinematic motion parameters of live organisms and rendering the corresponding swimming patterns on their robotic counterparts. For example, data from live lampreys have been used to reproduce anguilliform swimming gait in a robotic lamprey (Hultmark et al., 2007). Instead of manually tuning, using pre-defined kinematics, or focusing on a particular swimming pattern, we developed a unified scheme to learn such patterns, whereby the same scheme can be optimized for various patterns exhibited by different organisms.

Central pattern generators (CPGs) have long been studied as circuits in spinal cords of vertebrates for rhythmogenesis, that is, they are responsible for rhythmic motion generation without requiring periodic input (Ijspeert, 2008/05). They have been extensively used for robots inspired by nature (Niu et al., 2014; Crespi and Ijspeert, 2008; Zheng et al., 2022; Zhang et al., 2022) for motion generation. They receive commands from the midbrain for gait modulation and transition (Ijspeert et al., 2007/03). Inspired by this two-level gait generation and modulation process in nature, a similar bio-inspired scheme was conceived to explore the prospect of developing different motion patterns exhibited by different swimmers in the BCF category. CPGs are used for low-level gait generation. This requires an efficient high-level learning and optimization technique that can exploit and modulate CPGs to acquire diverse behaviors. Thus, we chose deep reinforcement learning (DRL).

DRL combines reward-based optimization using multi-layer perceptron models (Mnih et al., 2015/02). Its potential for use in various biomimetic platforms has been demonstrated (Zheng et al., 2022; Zhang et al., 2022; Yan et al., 2021). It has multiple variants; in this study, trust region policy optimization (TRPO) (Schulman et al., 2015) was used. Using DRL as high-level controller allows a robot to modulate the low-level controller by training without any supervised setup. The proposed schematic offers a learning framework for developing not one but a range of gaits. Hence, it has wide applicability on different platforms resembling the BCF anatomy.

1.3. Contributions of this paper

We coupled the DRL method to the backend of the CPGs (Fig. 3). This helped the robot to learn and explore multiple swimming gaits during the simulation. These gaits were validated and compared using the hardware of body and caudal fin robot (BCFbot) (Fig. 2). We designed the BCFbot maintaining the general anatomy of BCF swimmers, with a head, tail, and caudal fin at the posterior end. A generic design was opted such that it can exhibit multiple motion profiles to exploit their benefits. The contributions of this study are summarized as follows:

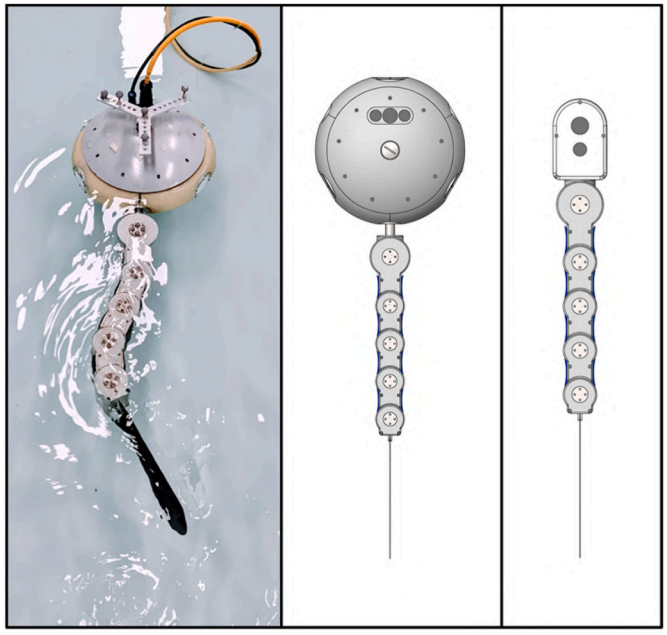

Fig. 2. *Left*: Snapshot of physical BCFbot while swimming in laboratory pool. *Center and Right*: CAD models of the robot in top view with a big head and a small head.

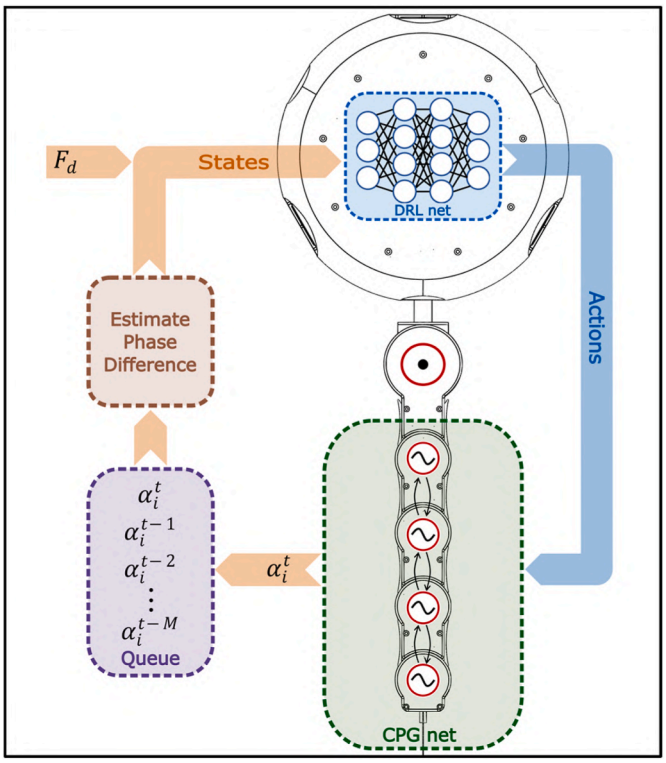

Fig. 3. Control Architecture overlayed on the sketch of the robot in the background. Empty circles in the DRL-net indicate neurons. Circles with sinusoids inside in the CPG-net indicate oscillators coupled by phase coupling indicated by small black arrows between them. α_i and F_d indicate the actual joint angle and the desired force.

1) A bio-inspired learning framework combining DRL and CPGs was proposed, from which a variety of swimming gaits resembling those of different natural BCF swimmers emerged.
2) Swimming gaits were visualized on a physical BCFbot. The multi-segment tail of the robot could imitate more than one gait in the same design making the visualization and differentiation of different gaits easier.
3) A thorough comparison was made between the swimming gaits which revealed certain benefits associated with each gait type making them appropriate for different scenarios and applications.
4) Features of the combined DRL-CPG approach were demonstrated, which are not possible using the DRL-only approach.

When a thrust target was given to the proposed DRL-CPG scheme, it generated a swimming gait pattern. The mechanism by which different gaits were developed involved setting several targets for the thrust generated by the robot. If attempted using a standard DRL method, it may simply attempt to increase the speed of actuation to deliver different thrust targets which may have some drawbacks. First, it can result in irregular motion due to a decrease in the resolution of data points. Second, different gaits will not develop because a higher thrust target will be achieved by increasing the actuation speed.

In the proposed scheme, motion generation and modulation are dealt with separately; tail-beat frequency can be fixed, and different swimming gaits can be explored based on different thrust targets. Once the gaits have been developed, the frequency can be easily modified while maintaining any particular gait. This gives complete control of the speed of the robot and allows comparison between the gaits. Moreover, the robot can shift smoothly from one gait to another because of the smooth convergence properties of the oscillators in short transient period. This framework eliminates the requirement for domain randomization, which is generally required in DRL-only cases for the transfer of policies from simulation to real scenarios.

The pairing of DRL with other methods can expand its ability to develop intelligent policies for learning and performing high-level tasks. In this study, we explored its potential for gaits development for a marine robot. The idea draws inspiration from two-stage gait generation, adaptation, and modulation in nature (Ijspeert et al., 2007/03). Setting up a framework by arranging the two components (DRL and CPG) in a hierarchical fashion and giving different set points allows the exploration of different ways to achieve those targets, thereby resulting in multiple useful gaits for different scenarios. This study investigated the framework on a BCF-type robot to explore three motion patterns (i.e., anguilliform, sub-carangiform, and carangiform).

The rest of the paper is organized as follows. Session II presents some relevant literature results, based on which Session III introduces our proposed learning method and training and control strategy. Session VI describes our testing rig setting up. The detailed pattern training results are shown in Session V with comparative studies in Session VI. A conclusion remark is given in Session VII.

2. Previous relevant work

Early efforts provide a basic categorization (Breder, 1926) based on which many later studies have been developed. A comprehensive insight (Lindsey, 1978) of aquatic locomotion modes present the BCF mode of locomotion as one of the main categories. Previously, head and tail amplitude have been considered as the main differentiating factors between the members of this category (Sfakiotakis et al., 1999). A recent study (Di Santo et al., 2021) suggests wavelength to be a better metric to classify the locomotion modes and that the BCF spectrum should better be considered as a continuum rather than distinctive classes.

Reproducing motion behaviors of organisms on robotic platforms allows researchers to study those organisms. It is difficult to conduct exhaustive studies on live animals. Robotic platforms provide repeatability and ease of experimentation (Gravish and Lauder, 2018). Practice started with big swimming machines (Morgansen et al., 2002; Triantafyllou and Triantafyllou, 1995; Saimek and Li, 2004). With miniaturization in actuation and energy storage, small-sized mobile marine robots can now be developed (Wang et al., 2022).

A review on mechanics of anguilliform gait, development of locomotion controllers and trajectory tracking of an eel-like robot can be referred to (Ostrowski and Burdick, 1998) and (McIsaac and Ostrowski, 2003). Anguilliform motion pattern is optimized on a multi-joint snake like robot (Crespi and Ijspeert, 2008) and on a robotic fish (Niu et al., 2013) for both forward and backward locomotion. The anguilliform pattern is difficult to mimic since it requires large number of DoF (Raj and Thakur, 2016). Whereas sub-carangiform or carangiform patterns are relatively easy to mimic. Hence, in the sub-carangiform and carangiform category we find quite a few efforts. For example, sub-carangiform pattern is exhibited in (Salumä et al., 2013) (Daou et al., 2011) with a single actuator and a complaint tail and in (Wu et al., 2014) with 4 actuators. The results in (Clapham and Hu, 2014; Farideddin et al., 2015; Low et al., 2010; EunJung and Youngil, 2004) are just a few examples of robots mimicking the carangiform pattern.

There are not many works that focus on achieving multiple motion patterns on a single platform. In (Fujiwara and Yamaguchi, 2017), sub-carangiform and carangiform patterns are realized but by making changes in design (adjusting compliance). In (Li et al., 2011), three patterns are realized by selecting corresponding kinematics from literature. Our focus is to realize different motion patterns on the same platform not through any change in design or rendering pre-determined kinematics but via learning different swimming gaits.

Biomimetic robots require periodic inputs and CPGs are widely used for this purpose (Ijspeert, 2008). (Niu et al., 2014; Crespi and Ijspeert, 2008; Zhang et al., 2022; Wu et al., 2014) are a few examples of their usage for BCF type robots. Its parameters have also been optimized using techniques like particle swarm optimization (Yu et al., 2016; Wang

et al., 2019) to maximize speed.

DRL has garnered traction over the years for its state-of-the-art optimization potential using neural networks. It can accommodate both discrete (Mnih et al., 2015) and continuous action spaces (Gao et al., 2021). Ranging from simple differential drives to bipeds and quadrupeds, it has applications in both terrestrial and marine domain (Yao et al., 2023), for example, propeller operated boat (Marchesini et al., 2021), and biomimetic fish-like robot (Yan et al., 2021). DRL has also been used along with oscillators. Such scheme has been applied to a hopper (Campanaro et al., 2021), a robotic salamander (Cho et al., 2019) for optimization of oscillators in simulation, and a soft robotic snake (Liu et al., 2023) to optimize locomotion controllers. In (Yan et al., 2021), a baseline behavior obtained by oscillators, is optimized using DRL for a multi-joint robotic fish. In (Yu et al., 2021), DRL is used on a robotic fish for target tracking using visual feedback. Similarly in (Zheng et al., 2022), it is used for attitude holding of a biomimetic fish in unknown flow fields. In (MiladShafiee, 2024), quadrupeds are trained using by combining DRL and CPGs.

In this study, we also optimize oscillators in a learning framework. However, instead of obtaining and optimizing one single locomotion gait, we obtain multiple motion patterns. The resultant setup can essentially obtain a continuum of motion patterns like the BCF spectrum observed among marine organisms. The framework adds up the benefits of DRL and CPGs such as easy modulation of output signals, guaranteed limit cycle behavior and easy transfer of policies from simulation to hardware. It should be the first time to achieve optimization of motion patterns using DRL and CPGs in a single marine robot with experimental validation.

3. The proposed learning and training methods

This session is to present the new proposed learning method with detailed algorithms and control scheme.

3.1. Deep reinforcement learning

A DRL framework is established in which a simulated BCFbot is trained in an episodic format. The agent takes an action $a(t)$ according to policy π i.e., $a(t) \sim \pi(s(t))$, shifts from state $s(t)$ to $s(t+1)$ and is rewarded r based on a preset reward definition. This transition takes place according to state transition dynamics $D = p(s(t+1)|s(t),a(t))$. An episode completes when this transition happens for T timesteps. Collective reward of an episode is called a Return. Q_π is the expected value of return upon taking action $a(t)$ in state $s(t)$ following the policy π as described above,

$$Q_\pi(s,a) = \mathbb{E}_{s(t+1),a(t+1)\ldots}\left[\sum_{t=0}^{T}\gamma^t r(s(t),a(t))\right] \tag{1}$$

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s). \tag{2}$$

The parameter γ is a limiting factor for future rewards. V_π is the expected return and A_π is called the advantage function.

Trust region policy optimization (TRPO) (Schulman et al., 2015) is adopted in which the policy π has a set of parameters θ and the following objective is maximized,

$$\underset{\theta}{\text{maximize}}\ \mathbb{E}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A_{\pi_{\theta_k}}(s(t),a(t))\right], \tag{3}$$

where π_{θ_k} is the policy which is being improved upon π_θ. TRPO takes a guaranteed step (3) to improve the policy from the previous one by measuring the Kullback-Leibler divergence, D_{KL}, between the old and the new policy,

$$\mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[D_{KL}\left(\pi_{\theta_k}(\bullet|s),\pi_\theta(\bullet|s)\right)\right] \le \delta, \tag{4}$$

which states that the divergence must be kept under a fixed value δ. Both the objective and the constraint are non-linear. For the main objective (3), linear approximation is done and for the constraint (4) quadratic approximation is done using Taylor expansion. Denoting the objective in (3) by $\mathscr{L}_{\theta_k}$, we can approximate (3) and (4) as,

$$\mathscr{L}_{\theta_k}(\theta) \approx \mathscr{L}_{\theta_k}(\theta_k) + g^T(\theta-\theta_k) \tag{5a}$$

$$D_{KL} \approx \frac{1}{2}(\theta-\theta_k)^T H(\theta-\theta_k), \tag{5b}$$

g being the first derivative of $\mathscr{L}_{\theta_k}$ and H being the second derivative of D_{KL},

$$g = \nabla_\theta \mathscr{L}_{\theta_k} = \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[\nabla_\theta \log\ \pi_\theta(a|s)\, A_{\pi_{\theta_k}}\right] \tag{6a}$$

$$H = \nabla_\theta^2 D_{KL}. \tag{6b}$$

Algorithm I. Pseudocode for Training

1. Initialize policy π_θ with random parameter set θ.
2. **for** $1 \le k \le K$ **do**
3. **for** $1 \le l \le L$ **do**
4. **for** $1 \le t \le T$ **do**
5. Sample an action $a_t \sim \pi_\theta$
6. Input the action a_t to the CPG-net to obtain inputs α_i^t for the agent by integrating system (9),
$$\alpha_i^t = u_i^t \sin(\phi_i^t) + d_i^t$$
7. Render α_i^t to the agent and record the next state s_{t+1} and the reward r_t.
8. **end for**
9. Store the transition in a set of trajectories $\mathcal{B}$.
10. **end for**
11. Calculate returns G_t and advantage estimates A_t.
12. Compute policy gradient g_k,
$$g_k = \frac{1}{L}\sum_{\mathcal{B}}\sum_{t=0}^{T}\nabla_\theta \log\pi_\theta(a|s)\, A_t$$
13. Compute the following relation,
$$x_k \approx H_k^{-1} g_k$$
H_k being the Hessian of the Kullback Liebler divergence (5b) between the old and new policies.
14. Compute step size,
$$\beta \approx \sqrt{\frac{2\delta}{x_k^T H_k x_k}}\, x_k$$
15. Update the policy parameters,
$$\theta_{k+1} = \theta_k + \eta\beta$$
η being the smallest value that satisfies the KL divergence constraint.
16. **end for**

Considering the above approximations, the original objective and the constraint can now be combined as,

$$\theta_{k+1} = \underset{\theta}{\text{argmax}}\quad g^T(\theta-\theta_k) \tag{7}$$

$$\text{s.t.}\quad \frac{1}{2}(\theta-\theta_k)^T H(\theta-\theta_k) \le \delta$$

which can be analytically solved as,

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}}\, H^{-1} g. \tag{8}$$

3.2. Rhythmogenesis

Rhythmogenesis is handled using CPGs. CPGs are implemented as a network of coupled oscillators. The green block in Fig. 3 represents the CPG network in which the red circles represent oscillators for individual joints. The arrows between the oscillators indicate phase couplings.

$$\dot{\phi}_i^t = 2\pi f + \zeta_i^t \tag{9a}$$

$$\zeta_i^t = \sum_j u_j^{t-1} c_{ij} \sin\left(\dot{\phi}_j^{t-1} - \dot{\phi}_i^{t-1} - \psi_{ij}\right) \tag{9b}$$

$$\ddot{u}_i^t = c_1\left[\frac{c_1}{4}\left(U_i - u_i^{t-1}\right) - \dot{u}_i^{t-1}\right] \tag{9c}$$

$$\ddot{d}_i^t = c_2\left[\frac{c_2}{4}\left(D_i - d_i^{t-1}\right) - \dot{d}_i^{t-1}\right] \tag{9d}$$

$$\alpha_i^t = u_i^t \sin\left(\phi_i^t\right) + d_i^t \tag{9e}$$

For each oscillator i in the network, u_i represents the amplitude while d_i and ϕ_i denote deviation and phase respectively. α_i is the actual output of the oscillator i. c_{ij} and ψ_{ij} specify couplings among the oscillators. The frequency f is kept constant during training and can be varied as required during testing. Constants are mentioned in Table II.

3.3. Control schematics and algorithm

In the schematics (Fig. 3), DRL part is implemented as a multilayer perceptron (MLP) with 2 layers (64 neurons per hidden layer). The parameters of this network are updated during training using TRPO (3) according to Algorithm I. This network outputs parameters of CPG network namely the phase couplings – ψ_{ij}. The oscillator network is implemented as a system of equations represented by the system in (9). The system (9a to 9e) is implemented as a python class whose input is the desired frequency, amplitude and phase difference between the rhythmic patterns. And the system outputs an array containing time evolution of the rhythmic patterns. In TRPO, multiple batches are collected before the policy is updated. During the batch collection step (lines 3 to 10 of Algorithm I), first the MLP is inferred using the latest available states. In our case, the action space of DRL is not the final action space since the policy output is not directly rendered on the robot. The architecture resembles (Hameed et al., 2022) in which the policy output (action space) is not directly rendered on the robot. The action space corresponds to the weights and goals of so-called dynamic motion primitives which are integrated to calculate final inputs for the robot joints.

In this case, the action space is set point for the CPG-net. With the latest available set point, oscillators are integrated to acquire actions for the robot joints. These actions are rendered on the robot in simulation which in turn returns the updated joint positions of the robot. The updated joint configurations are then saved in a queue along with a history of the last tail-beat cycle. This cycle is then analyzed to estimate phase difference between the joints. The updated information is then concatenated with the desired force and fed back to the MLP, and the loop goes on for the length of episode. Once the required number of batches are collected, the policy is updated according to update rules from lines 11 to 15 of Algorithm I.

Table 1
Cost of transport and turning.

Frequency (Hz)	$f = 0.7$	$f = 1.0$	$f = 1.3$	$f = 1.6$
Motion Pattern	**Cost of Transport-CoT ($\times 10^{-2}$ J/m)**			
Anguilliform	1.29	1.16	1.17	1.23
Sub-Carangiform	**0.88**	**0.73**	**0.71**	**0.67**
Carangiform	1.15	0.85	0.82	0.77
Motion Pattern	**Cost of Turning-CoTu ($\times 10^{-2}$ J/rad)**			
Anguilliform	0.61	0.58	0.53	0.96
Sub-Carangiform	**0.32**	**0.30**	**0.28**	**0.32**
Carangiform	0.51	0.47	0.44	0.46

Table 2
Values of constants.

γ	0.99	f	1.0 Hz[a]
δ	0.001	U_i	0.3 rad[a]
c_2	$c_1/4$		

[a] Held constant during training. Can be changed as required during testing.

3.4. Training methodology

A large amount of data is required to train models in DRL. The robot hardware cannot be used to collect that much data. Hence, a physics emulator (Todorov et al., 2012) is used to simulate the agent and to perform training. In the emulator a physical model of the robot is developed. The head is modelled as an ellipsoid, links as rectangular bars and joints as cylinders. The model can be seen in the supplementary video. Actuators are defined at joint locations. The emulator returns the updated state of the robot when the input α_i generated by Algorithm I is given to the actuators. This data is collected and is used to train the networks.

Supplementary data related to this article can be found online at https://doi.org/10.1016/j.oceaneng.2025.120322

During training the translational degrees of freedom (i.e., along x and y axes) are locked and therefore, the robot stays in its place. A force sensor is installed in the global frame of reference. The forces developed by the agent are recorded by the force sensor. This force has two components. The component along the longitudinal axis – F_x is the thrust or propulsive component since it is primarily responsible to propel the robot. F_y is the lateral component perpendicular to the thrust component (Fig. 4-right).

In the reward function, the difference between the actual and the desired force is minimized,

$$r = -\|F_d - F_a\|^2 \tag{10a}$$

$$\text{where} F_a = \sum_{i=n}^{m} \sqrt{{F_{x_i}}^2 + {F_{y_i}}^2} \tag{10b}$$

Here F_d represents desired set point for the force whereas F_a is the actual force recorded by the sensor averaged over $(m-n)$ timesteps. When the tail undulates, the joints move according to travelling wave profile and return to the original configuration after one tail-beat cycle. The duration of the cycle depends on tail-beat frequency. To generate a higher thrust force (F_x) at fixed frequency, swimming pattern must change. This change will also alter F_y since both components are coupled and one cannot change completely independent of the other. Using one

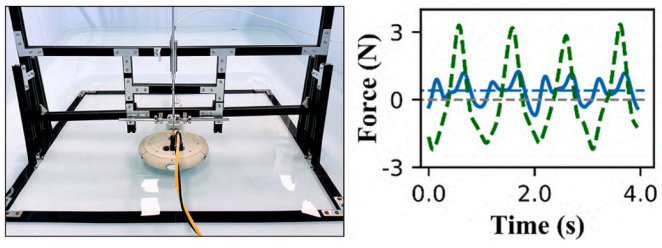

Fig. 4. *Left*: Thrust measurement setup. *Right*: Periodic force patterns developed by the robot while swimming with the anguilliform pattern at 1.0 Hz for 4 s. Blue (solid) line represents F_x (average ~0.4 N) and green (dashed) line represents F_y which oscillates around zero. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

component provides little room for exploration but using both components provides large room for exploration in the parameter space and therefore helps the development of different patterns.

It should be noted here that the average thrust can be controlled directly by controlling the frequency of oscillations. This method is commonly used in literature (Zhu et al., 2019/09) to control the thrust and the speed of such robots. The goal here is to develop different motion patterns. To avoid learning frequency to thrust relationship, the frequency of oscillations is kept constant throughout the training process. In this way the control architecture has to explore other means – different swimming gaits to deliver different target presets for the force. This is the key idea that enables the robot to learn and adopt multiple motion patterns.

To develop varied patterns, F_y is used in addition to F_x. It can be observed from Fig. 4-right that for each tail-beat cycle, the average of F_y remains zero for straight line motion contrary to F_x, average of which has to be non-zero for forward motion. At low frequencies, the effect of F_y is visible in the form sway of body about the longitudinal axis. But at high frequencies, this effect is not very visible. In real fishes, this effect is also mitigated by the flat shape of the fishes posing high drag resistance perpendicular to the longitudinal axis. Therefore, F_y is also critical in determining the swimming gait, and hence, it is included in the reward function.

4. Experimental setup

4.1. Robot hardware

The robot design (Fig. 2) is inspired by BCF swimmers. The head contains a Raspberry Pi (RPi 4B) microcontroller, and a battery. An external laptop communicates with RPi to send and receive high-level commands. The body part is composed of four 1-DoF joints. The distance between two consecutive joints is 6.5 cm. These segments (and the head) are made with machined plastic and have position servos inside to actuate them. At the end of the tail, there is a caudal fin. The caudal fin is carved out of a 0.7 mm carbon fibre sheet.

4.2. Experimental setup

A small laboratory swimming pool (2.5 m × 4 m) (water level ~ 1 ft) is used to perform experiments. The buoyancy of the robot is set to keep the robot's body (tail and fin) just under the surface of water. On the corners of the pool, four tracking cameras are installed at 3 m height. The setup can track the robot while it swims in the pool by tracking the infrared reflective markers installed on the head of the robot.

For thrust measurement, a static structure is assembled inside the pool (Fig. 4-left). A rod is attached to the robot's head. Two bearings are used to let the rod rotate around x and y axes passing through the center of rod. The bearings are affixed to the static structure, due to which the rod and the robot itself cannot translate in any direction. The other end of the rod meets a load cell which measures the forces developed by the robot. Data from load cells is first amplified and then recorded using a data recorder. A camera is installed on the top to capture waveforms induced in the robot's body.

5. Pattern training results

To visualize the motion patterns, we remove the restriction previously imposed during training and let the agent swim. We will first observe static postures of the agent (Figs. 5a and 6a) and then the trajectories traced by the fin of the agent while it swims as shown by the periodic patterns in Figs. 5b and 6b. It provides a good way to visualize and compare swimming gaits. Finally, we record videos for both simulation and hardware results which can be viewed in the supplementary material.

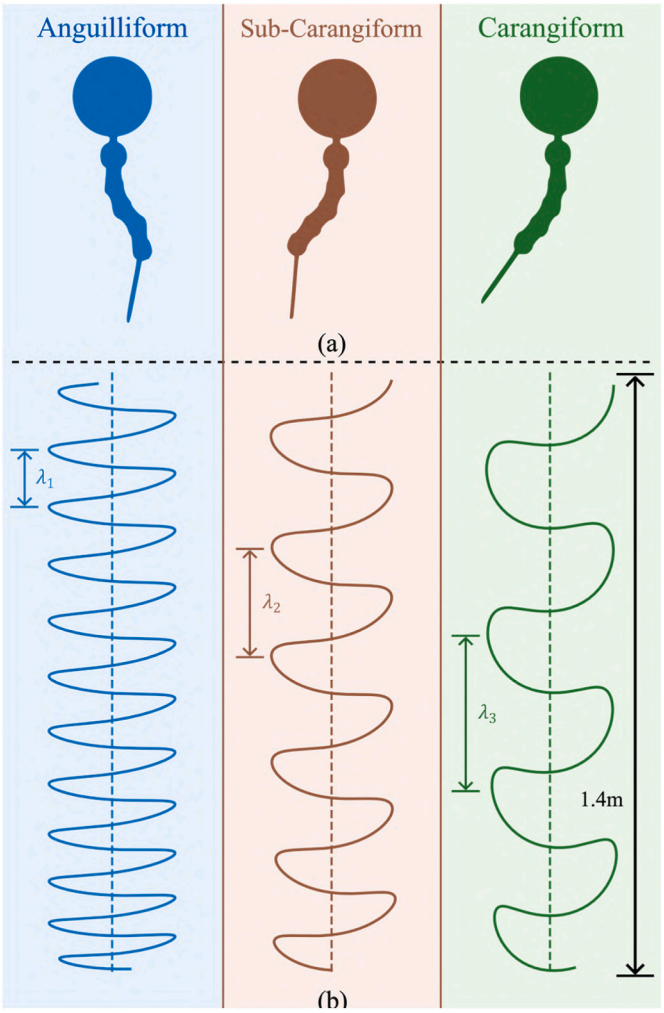

Fig. 5. (a) Snapshots of the robot in simulation when the framework renders the anguilliform (left), sub-carangiform (center), and carangiform (right) motion patterns after training. (b) Fin tracks traced by the fin of the robot in simulation while it swims with the patterns shown in (a).

5.1. Simulation results

The three results from the left to right in Fig. 5 are corresponding to the force targets (F_d) from 1.8 N, 3.0 N, to 4.2 N respectively. For the lowest target (1.8 N), the motion pattern in Fig. 5a (left) looks similar to the anguilliform pattern shown in Fig. 1. The wavenumber is the largest as almost half of the propulsive wave is contained in the agent's body. As the set point for the desired force is increased by 1.2 N, the resemblance of tail configuration shifts to the sub-carangiform and eventually to the carangiform pattern on further increase of 1.2 N. These poses should be observed along with Fig. 1. The agent now contains almost one quarter (Fig. 5a-center) and even less than one quarter (Fig. 5a-right) of the travelling wave within its body.

Fig. 5b shows fin trajectories for the three cases. The wavelength (λ_1 = 12.2 cm) is the shortest in the anguilliform pattern. The waveform starts to expand, and the wavelength increases to 24.5 cm in the sub-carangiform pattern and finally to 38 cm in the carangiform case. Therefore, as the desired force targets increase, the wavelengths increase. The ratio of the wavelengths from the anguilliform to sub-carangiform pattern is 0.5 and from the sub-carangiform to carangiform pattern is 0.6. Later, these ratios will be compared with the hardware results.

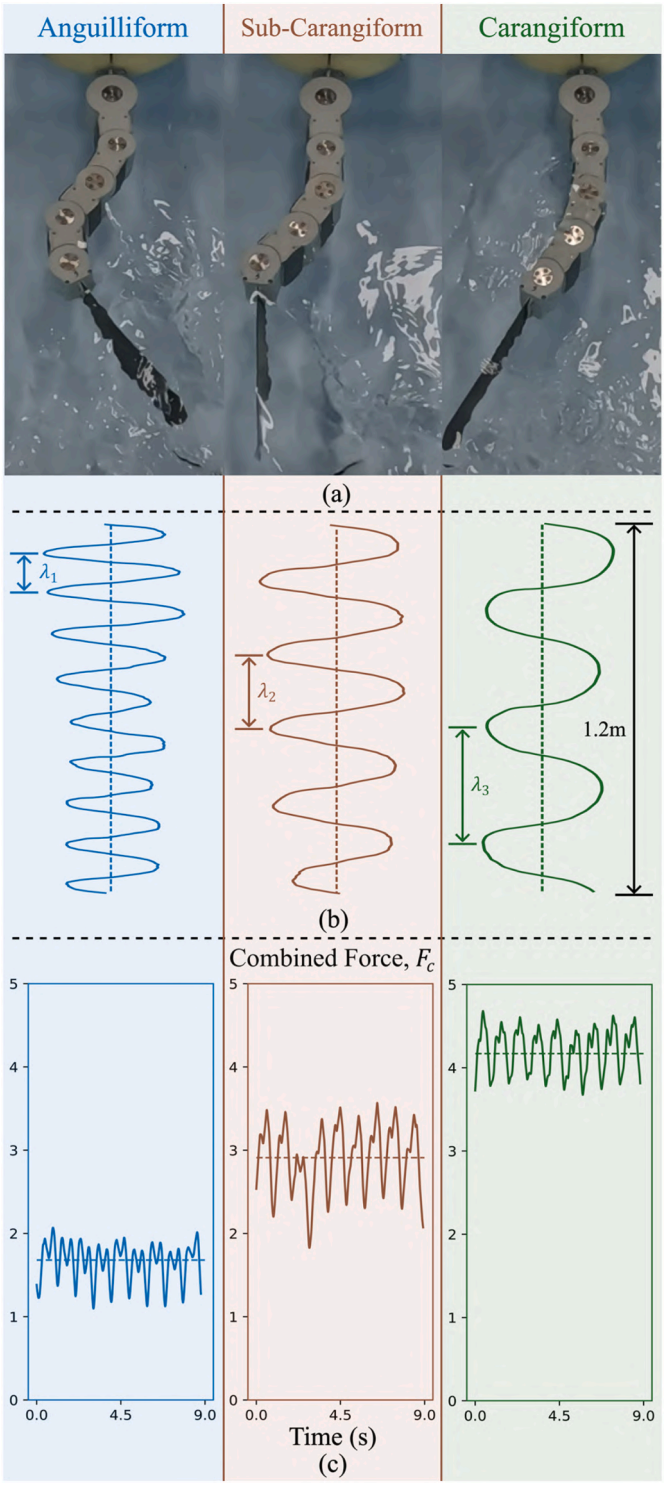

Fig. 6. (a) Snapshots of tail when the anguilliform (*left*), sub-carangiform (*center*) and carangiform (*right*) patterns are rendered on the robot hardware while it is attached to the thrust measurement setup. (b) Fin trajectories traced by the fin of the robot while swimming with the patterns shown in (a). (c) Combined force output F_c (N) of the robot while swimming with the three patterns in (a).

5.2. Testing results

Hardware results are summarized in Fig. 6. The pose on the left (of Fig. 6a) is the anguilliform pattern. It has similar body and fin configuration as the one in simulation (Fig. 5a-left) whereby the tail of the robot contains one half of the travelling wave. The pose in the center (of Fig. 6a), corresponds to the sub-carangiform pattern and is similar to the body posture in Fig. 5a-center. Lastly, the tail configuration on the right (of Fig. 6a) looks similar to the carangiform pattern and resembles Fig. 5a-right. The hardware poses in all three cases conform with the simulation results. One can observe the similarity of hardware poses (Fig. 6a) and simulation poses (Fig. 5a) with the classical BCF modes introduced in Sec. I (Fig. 1).

Next, fin trajectories are compared between the simulation (Fig. 5b) and hardware experiments (Fig. 6b). The one on the left is acquired when the robot swims with the anguilliform pattern whereby the wavelength is the shortest (13 cm) among all. It starts to expand when it swims with the sub-carangiform pattern (24 cm) and becomes the largest in the case of carangiform pattern (40 cm) thereby giving the ratios of λ_1 and λ_2 to be 0.54 and from λ_2 to λ_3 to be 0.6. These ratios match closely with the ones in simulation (Sec. VIA).

Force measurement results on the hardware are shared in Fig. 6c. These results are not individual component of forces, but the combined force of both components calculated by $F_c = \left(F_x^2 + F_y^2\right)^{1/2}$, same as the definition of F_a in (10b). Combined force (F_c) is shown by solid and its average ($\overline{F_c}$) by dashed lines. For the anguilliform case, the average is around 1.7 N. It rises to 2.9 N and 4.1 N for the sub-carangiform and carangiform cases. There exists a clear difference in the combined average forces in the three cases which makes it easier for the control framework to explore multiple gaits in the parameter space.

Force components for the anguilliform case are shown in Fig. 4 and for the sub-carangiform and carangiform cases in Fig. 7. The nature of profiles is characteristic to each swimming gait. For the first two cases, the rise or fall in F_y is gradual whereas for the last case, it stays at maxima before falling rapidly to the next minima. The average value of F_x increases from the anguilliform to carangiform case and remains zero for F_y.

It should be noted from Figs. 5b and 6b that the fin trajectories expand from left to right. It can also expand if the tail-beat frequency is increased. If the frequency is increased, the body will wriggle faster, and the agent will swim faster resulting in an expanded fin track compared to the one generated at low speed when observed for a fixed distance of travel. But the expansion phenomena shown in Figs. 5b and 6b is not due to any change in tail-beat frequency. The frequency is kept constant (at 1.0 Hz) in all three cases. The presence of 9 cycles in the time duration of 9 s (Fig. 6c) also confirms that the tail-beat frequency remains same in all the cases. Hence, the expansion and the increase in force is purely due to changes in swimming gaits. The tail-beat frequency is held constant by the oscillator network. Therefore, to cope up with increasing force targets, the DRL part must evolve the swimming gait to generate larger forces and in doing so, it develops different swimming gaits.

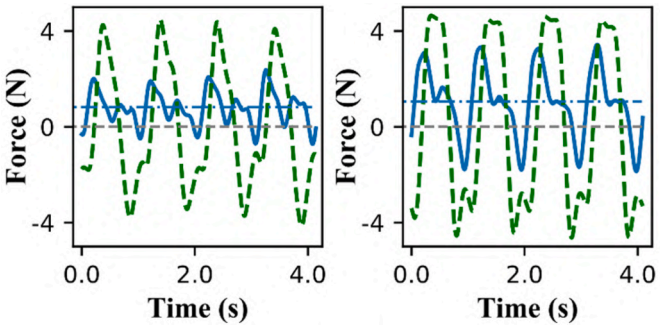

Fig. 7. Periodic force patterns developed by the robot while swimming with the sub-carangiform (*left*) and carangiform (*right*) gaits at 1.0 Hz for 4 s. Blue (solid) line represents the thrust component – F_x whereas green (dashed) line represents the y-component – F_y which oscillates around zero. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

6. Comparison of swimming performance between different swimming gaits

In this part experiments are performed to compare the gaits with each other in terms of swimming speed and cost of transport. Frequency is a key parameter to control the speed of such biomimetic robots. Using DRL-only scheme, it is not straightforward to control the speed on biomimetic platforms since it involves a network that has to be queried at every iteration. In the proposed framework, since the rhythmic pattern generation is handled separately, it is possible to control and vary the key parameters of rhythmic motion e.g., the period of oscillations can be varied by altering the respective parameter in the system (9) of oscillators.

6.1. Straight-line swimming

The speed of the robot at different frequencies while swimming with the three patterns is compared in Fig. 8 (left). Overall, we observe the anguilliform pattern to be the worst performing with speeds ranging from 12 cm/s to 17 cm/s. The sub-carangiform gait follows, giving a minimum of 22.5 cm/s at 0.7 Hz and a maximum of 36 cm/s at 1.6 Hz. The carangiform pattern performs the best with outputs ranging from 30 cm/s to 45 cm/s.

Cost of transport (CoT) (Zhu et al., 2019/09) is a parameter that can be helpful to compare the patterns,

$$CoT = P_{in}/v \tag{11}$$

where P_{in} is the power consumption of the robot while swimming with linear speed v. Table I shares CoT values for all the cases. CoT decreases with increasing tail-beat frequency. The anguilliform mode is the most expensive owing to its poor linear speed performance whereas the sub-carangiform pattern is the most economical among all giving the lowest value of 0.67 J/m at 1.6 Hz.

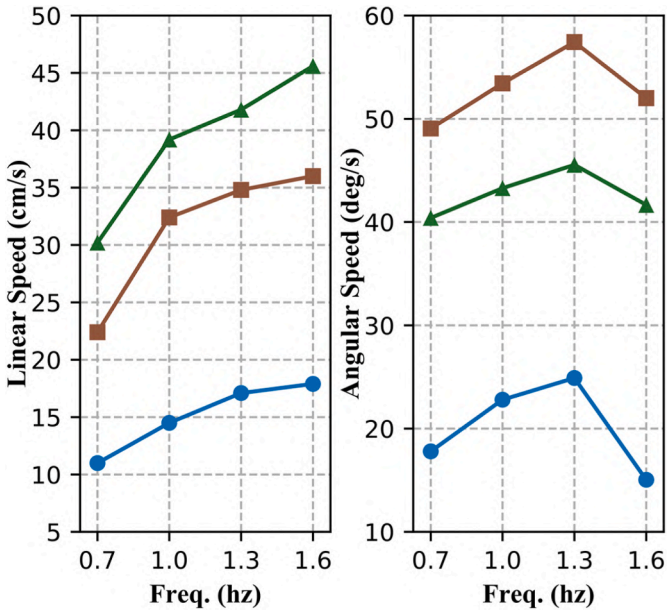

Fig. 8. Linear and angular speed comparison of the robot while swimming with different motion patterns at four different tail-beat frequencies. Blue, brown, and green markers represent the robot's linear (*left graph*) and angular (*right graph*) speeds while swimming with the anguilliform, sub-carangiform and carangiform modes respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

6.2. Turning

To quantify turning performance, non-zero constant deviation (D_i = [0.0, 0.3, 0.3, 0.3] rad) (9d) is input to CPG network so that the robot can perform turning motion. Fig. 8 (right) summarizes the turning performance in terms of turning rates observed at 4 different frequencies while the agent employs the three gaits.

Unlike straight-line swimming speed results, sub-carangiform gait performs the best in the turning case with a maximum turning rate of 57°/s at 1.3 Hz. The anguilliform pattern yields very low turning rates (maximum of 25°/s only). The carangiform pattern lies in the middle of two with steering rates ranging from 40°/s to 45°/s.

Unlike the linear motion case, turning at high frequency does not necessarily mean that it would be faster. In straight-line swimming, when the tail undulates, it generates propulsion for the entire tail-beat cycle. However, during turning, this continuity breaks. During turning, there is a pushing phase in which the robot uses its fin to push water to turn itself. Then, there is a retraction phase in which the tail retracts back. This creates an effect opposite to the desired turning direction. Lower frequency allows the robot enough time to make complete use of the pushing phase. At high frequency the retraction phase breaks the angular momentum. This is the reason behind the fall of angular turning rates after 1.3 Hz (Fig. 8-right). This effect is also observed and discussed in (Zhong et al., 2017).

We do not find any yardstick to rate turning performance on such platforms. Therefore, in the style of cost of transport (CoT), we calculate cost of turning ($CoTu$).

$$CoTu = P_{in}/\omega \tag{12}$$

where ω is the angular velocity of the robot. The results are shared in Table 1. The cost is minimum for the case of sub-carangiform mode and the anguilliform is the most expensive mode for turning as well.

6.3. Backward mode and free-swimming snapshots

Backward Mode. A distinguishing feature of some anguilliform swimmers is their ability to swim backward. This feature can be developed by setting a negative target for the desired force. In this case, a forward travelling wave (originating from the fin and terminating at the head) is generated thereby generating negative average thrust. The corresponding result can be observed in Fig. 9. A free-swimming experiment is also conducted for the backward mode which reports average speed of 8 cm/s (Fig. 9-right).

Free-Swimming Experiment. Snapshots from the free-swimming experiments are shown in Fig. 10. Fig. 10a, 10b and 10c represent the anguilliform, sub-carangiform, and carangiform cases respectively. Tail-

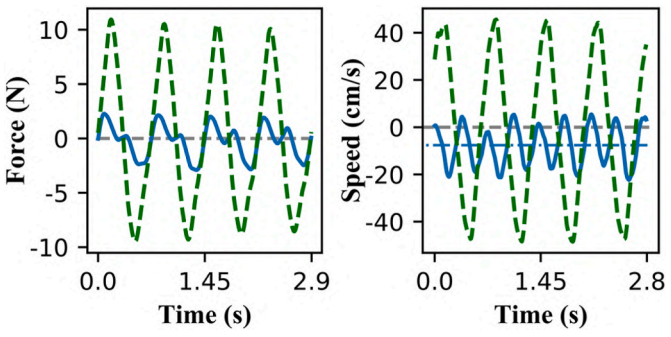

Fig. 9. *Left*: Thrust F_x (solid blue) and lateral F_y (dashed green) forces developed by the robot undulating in backward mode while attached to the thrust measurement setup. *Right*: Velocities V_x (solid blue) and V_y (dashed green) while the robot is swimming freely in backward mode. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

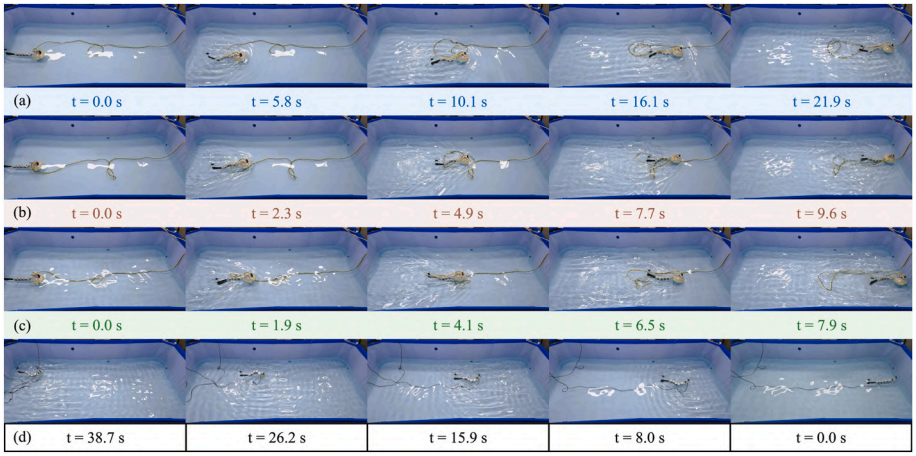

Fig. 10. Stills from free-swimming experiments for the anguilliform (a), sub-carangiform (b), carangiform (c), and backward (d) motion modes. Time stamps are mentioned at the bottom of snapshots. (a), (b), and (c) are from left to right whereas (d) is from right to left.

beat frequency in all three cases is the same (1.0 Hz) but due to different gaits, the linear speeds vary and the robot takes different time intervals to swim for the same distance.

The fourth sub-figure (Fig. 10d) is for the backward swimming case in which the robot starts from the right side, and swims in backward mode to the left side. In this mode, the fin is in the front of the robot. The motion pattern looks like the anguilliform pattern. The hardware that has been used so far has a big and heavy head and it is difficult for the robot to swim with a bulky posterior end. So, a smaller head (Fig. 2-right) is used in this case. The slow speed in this case is because there is no fin at the point where the travelling wave terminates.

7. Conclusions and discussions

A control schematic is proposed to train the robot in simulation to learn different swimming modes. These modes are then rendered on the physical robot and compared in terms of their performance. It is revealed that all of the developed swimming gaits have certain benefits as follows.

1) The carangiform mode produces the fastest linear speed, but the power consumption in this mode is also very high, which inhibits it to be the most economical mode. The sub-carangiform swimming gait for its low wattage consumption stands out in terms of economy. Although it is not the leading mode in terms of speed, it follows the leading mode closely.
2) Similarly, the predominance of undulatory feature in the sub-carangiform mode makes it an ideal candidate for turning as well. The carangiform pattern, which lies a step further towards the oscillatory end of the spectrum, makes it an ideal candidate for straight-line swimming but not for turning. This means that the turning phenomena is better achieved with a mixture of undulatory and oscillatory modes rather than pure modes.
3) The anguilliform mode performs worse among all the three modes in both linear and angular cases. But the backward swimming in this mode gives the robot a unique ability. Hence, all three modes have benefits of their own.

The classical kinematic classification of BCF swimmers with four major modes should be better considered as a continuum (Di Santo et al., 2021). This motivates the development of a flexible and generic control architecture for robots mimicking BCF anatomy to tap the benefits of not one but a range of organisms. An effort is made in this regard by combining DRL with CPG to develop multiple swimming patterns similar to ones observed among natural BCF swimmers for a single robot that mimics BCF anatomy.

In this work, it is not intended to generate a gait that surpasses results shared in other works in speed or any other aspect. Some swimming gaits are naturally slow, and some are naturally fast. In literature, for such robots, efforts are usually concentrated to performing navigation using a singular gait and not much attention is paid towards emergence and reproduction of multiple gaits. In this article, the focus is kept at producing a family of gaits and their comparison. Further comparisons fall beyond the scope of this investigation which pertains mostly to the fundamental point that is how we can obtain various patterns. The results in (Li et al., 2011) render three motion patterns on the same robotic fish but demonstrate slight difference in performance with averaged linear speeds of 17, 21, and 22 cm/s for the three cases. However, the results in this study report obvious difference and better performance quantitatively.

Rhythmogenic circuits have been studied in animals for a long time (Ijspeert, 2008/05; MacKay-Lyons, 2002). Central control system plays a role in modulating these circuits to acquire desired behavior (Ijspeert et al., 2007/03). It raises a question whether one central control system can generate patterns exhibited by different swimmers. To explore this aspect, a similar bioinspired scheme is realized whereby a central system commands pattern generators in the peripheral system to acquire diverse behavior. It turns out that the central network can learn to optimize and modulate the pattern generator to extract multi-modal behavior. Hence, the study realizes natural gait generation mechanism and provides a step to further study the biomechanics of gait generation in BCF type swimmers.

In this study, testing is done in a laboratory pool to visualize the gaits and to highlight basic differences between them such as the speed and cost of travel. Real world testing can better demonstrate the performance of different modes. For example, the impact of disturbance from waves and water currents on performance of different modes is an interesting direction to explore. It requires solving a few challenges such as communication. Wired communication is used in this study which is not very feasible in real scenarios. Similarly, external tracking setup is used to collect position and velocity information. Position tracking in marine environment is challenging. These are some limitations of this study that can be worked upon in a later study.

Future work will include tackling the above challenges. Also, we can add a navigation layer to this framework to compare navigation performance of different gaits.

CRediT authorship contribution statement

Imran Hameed: Writing – original draft, Methodology,

Investigation, Formal analysis, Data curation, Conceptualization. **Xu Chao:** Validation, Investigation, Data curation. **David Navarro-Alarcon:** Validation, Supervision, Data curation. **Xingjian Jing:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization.

Declaration of competing interest

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

Acknowledgements

The work is supported by a startup fund of City University of Hong Kong (9380140), Shenzhen-HK-Macau Scheme-C (9240115) and an Innovation and Technology Fund project of HK ITC (9443005); and all testing and prototypes were conducted in the Laboratory of Nonlinear Dynamics, Vibration, & Control, CityU (Xingjian JING is the corresponding author).

References

Barrett, D., Grosenbaugh, M., Triantafyllou, M., 1996. The optimal control of a flexible hull robotic undersea vehicle propelled by an oscillating foil. In: Proceedings of Symposium on Autonomous Underwater Vehicle Technology, pp. 1–9.
Breder, C.M., 1926. The locomotion of fishes. Zoologica : scientific contributions of the New York Zoological Society 4 (5), 159–297.
Campanaro, L., Gangapurwala, S., De Martini, D., Merkt, W., Havoutis, I., 2021. CPG-actor: reinforcement learning for central pattern generators. In: Towards Autonomous Robotic Systems. Springer International Publishing, Cham, pp. 25–35.
Cho, Y., Manzoor, S., Choi, Y., 2019. Adaptation to environmental change using reinforcement learning for robotic salamander. Intell. Serv. Robot. 12 (3), 209–218.
Clapham, R.J., Hu, H., 2014. iSplash-II: realizing fast carangiform swimming to outperform a real fish. In: *2014 IEEE/RSJ International Conference On Intelligent Robots And Systems*, 14-18 Sept, pp. 1080–1086.
Crespi, A., Ijspeert, A.J., 2008. Online optimization of swimming and crawling in an amphibious snake robot. IEEE Trans. Robot. 24 (1), 75–87.
D'AoUT and Aerts, 1999. A kinematic comparison of forward and backward swimming in the eel Anguilla anguilla. J. Exp. Biol. 202 (Pt 11), 1511–1521.
Daou, H.E., Salumäe, T., Ristolainen, A., Toming, G., Listak, M., Kruusmaa, M., 2011. A bio-mimetic design and control of a fish-like robot using compliant structures. In: 2011 15th International Conference on Advanced Robotics (ICAR), pp. 563–568.
Di Santo, V., et al., 2021. Convergence of undulatory swimming kinematics across a diversity of fishes. Proc. Natl. Acad. Sci. USA 118 (49), e2113206118, 2021/12/07.
EunJung, K., Youngil, Y., 2004. Design and dynamic analysis of fish robot: PoTuna. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, vol. 5, pp. 4887–4892, 26 April-1 May 2004 2004, vol. 5.
Farideddin Masoomi, S., Gutschmidt, S., Chen, X., Sellier, M., 2015. The kinematics and dynamics of undulatory motion of a tuna-mimetic robot. Int. J. Adv. Rob. Syst. 12 (7), 83.
Fras, J., Noh, Y., Macias, M., Wurdemann, H., Althoefer, K., 2018. Bio-inspired Octopus robot based on novel soft fluidic actuator. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1583–1588.
Fujiwara, S., Yamaguchi, S., 2017. Development of fishlike robot that imitates carangiform and subcarangiform swimming motions. J. Aero Aqua Bio-mech. 6, 1–8.
Gao, H., Xiao, X., Qiu, L., Meng, M.Q.H., King, N.K.K., Ren, H., 2021. Remote-Center-of-Motion recommendation toward brain needle intervention using deep reinforcement learning. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 8295–8301.
Gao, T., Lin, Y., Qiu, L., Ho Tse, Z.T., Ren, H., 2021. Effects of cross-flow fan on hydrodynamic and acoustic performance of underwater fan-wing thruster. Ocean. Eng. 241, 110078.
Gravish, N., Lauder, G.V., 2018. Robotics-inspired biology. J. Exp. Biol. 221 (7), jeb138438.
Hameed, I., Chao, X., Navarro-Alarcon, D., Jing, X., 2022. Training dynamic motion primitives using deep reinforcement learning to control a robotic tadpole. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6881–6887, 23-27 Oct. 2022.
Hultmark, M., Leftwich, M., Smits, A., 2007. Flowfield measurements in the wake of a robotic lamprey. Exp. Fluid 43, 683–690.
Ijspeert, A.J., 2008. Central pattern generators for locomotion control in animals and robots: a review. Neural Network. 21 (4), 642–653.
Ijspeert, A.J., Crespi, A., Ryczko, D., Cabelguen, J.-M., 2007. From swimming to walking with a salamander robot driven by a spinal cord model. Science 315 (5817), 1416–1420.
Junzhi, Y., Min, T., Shuo, W., Erkui, C., 2004. Development of a biomimetic robotic fish and its control algorithm. IEEE Trans. Sys. Man Cybern., Part B (Cybernetics) 34 (4), 1798–1810.
Katzschmann, R.K., DelPreto, J., MacCurdy, R., Rus, D., 2018. Exploration of underwater life with an acoustically controlled soft robotic fish. Sci. Robot. 3 (16), eaar3449.
Kopman, V., Porfiri, M., 2013. Design, modeling, and characterization of a miniature robotic fish for research and education in biomimetics and bioinspiration. IEEE ASME Trans. Mechatron. 18 (2), 471–483.
Li, W., Tianmiao, W., Guanhao, W., Jinlan, L., 2011. A novel method based on a force-feedback technique for the hydrodynamic investigation of kinematic effects on robotic fish. In: 2011 IEEE International Conference on Robotics and Automation, pp. 203–208.
Li, G., et al., 2021. Self-powered soft robot in the mariana trench. Nature 591 (7848), 66–71.
Lighthill, M.J., 1960. Note on the swimming of slender fish. J. Fluid Mech. 9 (2), 305–317.
Lindsey, C.C., 1978. Form, function, and locomotory habits in fish. Fish Physiol. 7, 1–100.
Liu, X., Onal, C.D., Fu, J., 2023. Reinforcement learning of CPG-regulated locomotion controller for a soft snake robot. IEEE Trans. Robot. 39 (5), 3382–3401.
Low, K.H., Chong, C.W., Zhou, C., 2010. Performance study of a fish robot propelled by a flexible caudal fin. In: 2010 IEEE International Conference on Robotics and Automation, pp. 90–95.
MacKay-Lyons, M., 2002. Central pattern generation of locomotion: a review of the evidence. Phys. Ther. 82 (1), 69–83.
Marchesini, E., Corsi, D., Farinelli, A., 2021. Benchmarking safe deep reinforcement learning in aquatic navigation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5590–5595, 27 Sept.-1 Oct. 2021.
McIsaac, K.A., Ostrowski, J.P., 2003. A framework for steering dynamic robotic locomotion systems. Int. J. Robot Res. 22 (2), 83–97.
Meng, Y., Wu, Z., Dong, H., Wang, J., Yu, J., 2022. Toward a novel robotic manta with unique pectoral fins. IEEE Trans. Sys. Man Cybern.: Systems 52 (3), 1663–1673.
Milad Shafiee, G.B.a.A.I., 2024. ManyQuadrupeds: learning a single locomotion policy for diverse quadruped robots. In: *IEEE International Conference On Robotics and Automation ICRA 2024*, Yokohama, Japan.
Mnih, V., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.
Morgansen, K.A., Vela, P.A., Burdick, J.W., 2002. Trajectory stabilization for a planar carangiform robot fish. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), vol. 1, pp. 756–762 vol. 1.
Niu, X., Xu, J., Ren, Q., Wang, Q., 2013. Locomotion generation and motion library design for an anguilliform robotic fish. Journal of Bionic Engineering 10 (3), 251–264.
Niu, X., Xu, J., Ren, Q., Wang, Q., 2014. Locomotion learning for an anguilliform robotic fish using central pattern generator approach. IEEE Trans. Ind. Electron. 61 (9), 4780–4787.
Ostrowski, J., Burdick, J., 1998. The geometric mechanics of undulatory robotic locomotion. Int. J. Robot Res. 17 (7), 683–701.
Raj, A., Thakur, A., 2016. Fish-inspired robots: design, sensing, actuation, and autonomy—a review of research. Bioinspiration Biomimetics 11 (3), 031001.
Saimek, S., Li, P.Y., 2004. Motion planning and control of a swimming machine. Int. J. Robot Res. 23 (1), 27–53.
Salumäe, T., Kruusmaa, M., 2013. Flow-relative control of an underwater robot. Proc. R. Soc. A 469 (2153), 20120671.
Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P., 2015. Trust region policy optimization. In: International Conference on Machine Learning, pp. 1889–1897.
Sfakiotakis, M., Lane, D.M., Davies, J.B.C., 1999. Review of fish swimming modes for aquatic locomotion. IEEE J. Ocean. Eng. 24 (2), 237–252.
Singh, H., Maksym, T., Wilkinson, J., Williams, G., 2017. Inexpensive, small AUVs for studying ice-covered polar environments. Sci. Robot. 2 (7), eaan4809.
Todorov, E., Erez, T., Tassa, Y., 2012. MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033.
Triantafyllou, M., Triantafyllou, G., 1995. An efficient swimming machine. Sci. Ame. - SCI AMER 272, 64–70.
Wang, W., Gu, D., Xie, G., 2019. Autonomous optimization of swimming gait in a fish robot with multiple onboard sensors. IEEE Trans. Sys. Man Cybern.: Systems 49 (5), 891–903.
Wang, R., Wang, S., Wang, Y., Cheng, L., Tan, M., 2022. Development and motion control of biomimetic underwater robots: a survey. IEEE Trans. Sys. Man Cybern.: Systems 52 (2), 833–844.
Wu, Z., Yu, J., Tan, M., Zhang, J., 2014. Kinematic comparison of forward and backward swimming and maneuvering in a self-propelled sub-carangiform robotic fish. Journal of Bionic Engineering 11 (2), 199–212.
Yan, Q., Han, Z., Zhang, S.-w., Yang, J., 2008. Parametric research of experiments on a carangiform robotic fish. Journal of Bionic Engineering 5 (2), 95–101.
Yan, S., Wu, Z., Wang, J., Tan, M., Yu, J., 2021. Efficient cooperative structured control for a multijoint biomimetic robotic fish. IEEE ASME Trans. Mechatron. 26 (5), 2506–2516.
Yao, Q., et al., 2023. Learning-based propulsion control for amphibious quadruped robots with dynamic adaptation to changing environment. IEEE Rob. Autom. Lett. 8 (12), 7889–7896.
Yu, J., Wu, Z., Wang, M., Tan, M., 2016. CPG network optimization for a biomimetic robotic fish via PSO. IEEE Transact. Neural Networks Learn. Syst. 27 (9), 1962–1968.
Yu, J., Wu, Z., Yang, X., Yang, Y., Zhang, P., 2021. Underwater target tracking control of an untethered robotic fish with a camera stabilizer. IEEE Trans. Sys. Man Cybern.: Systems 51 (10), 6523–6534.
Zhang, T., et al., 2022. From simulation to reality: a learning framework for fish-like robots to perform control tasks. IEEE Trans. Robot. 1–18.

Zheng, J., Zhang, T., Wang, C., Xiong, M., Xie, G., 2022. Learning for attitude holding of a robotic fish: an end-to-end approach with sim-to-real transfer. IEEE Trans. Robot. 38 (2), 1287–1303.
Zhong, Y., Li, Z., Du, R., 2017. A novel robot fish with wire-driven active body and compliant tail. IEEE ASME Trans. Mechatron. 22 (4), 1633–1643.
Zhu, J., White, C., Wainwright, D.K., Di Santo, V., Lauder, G.V., Bart-Smith, H., 2019. Tuna robotics: a high-frequency experimental platform exploring the performance space of swimming fishes. Sci. Robot. 4 (34), eaax4615.