IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Learning-based Propulsion Control for Amphibious Quadruped Robots with Dynamic Adaptation to Changing Environment

Qingfeng Yao[1], Linghan Meng[1], Qifeng Zhang[1], Jing Zhao[1], Joni Pajarinen[2], Xiaohui Wang[3], Zhibin Li[4], and Cong Wang[1]

***Abstract*—This paper proposes a learning-based adaptive propulsion control (APC) method for a quadruped robot integrated with thrusters in amphibious environments, allowing it to move efficiently in water while maintaining its ground locomotion capabilities. We designed the specific reinforcement learning method to train the neural network to perform the vector propulsion control. Our approach coordinates the legs and propeller, enabling the robot to achieve speed and trajectory tracking tasks in the presence of actuator failures and unknown disturbances. Our simulated validations of the robot in water demonstrate the effectiveness of the trained neural network to predict the disturbances and actuator failures based on historical information, showing that the framework is adaptable to changing environments and is suitable for use in dynamically changing situations. Our proposed approach is suited to the hardware augmentation of quadruped robots to create avenues in the field of amphibious robotics and expand the use of quadruped robots in various applications.**

***Index Terms*—Quadruped robots, amphibious robots, robot learning, reinforcement learning.**

I. Introduction

AMPHIBIOUS environments, with their diverse terrains and conditions, pose unique navigation challenges for robotic systems. Legged robots, due to their ability to leverage discrete contact points [1], have demonstrated enhanced performance across various terrains compared to other robot types. For instance, quadruped robots can adapt foot placement to accommodate diverse terrains and tasks. Nonetheless, despite control theory and reinforcement learning algorithms facilitating efficient locomotion across different terrains [2], the optimization of movements in high-dimensional action spaces remains a substantial challenge [3], particularly in amphibious environments.

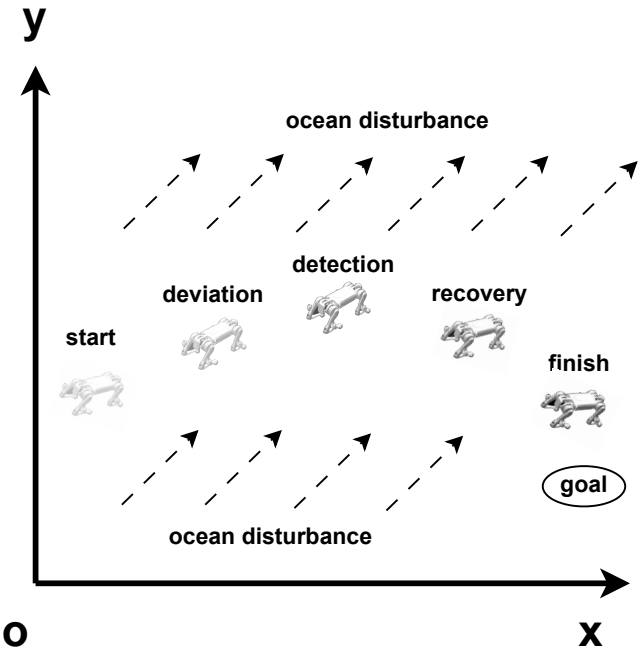

Fig. 1. Our model for adaptive propulsion control enables quadruped robots to compensate for water disturbance through real-time adjustments. Our approach can predict current disturbances, dynamically modify control actions, and rectify the perturbations.

Quadruped robots hold immense potential for amphibious exploration [4]. However, in scenarios involving floating conditions, the capabilities of quadruped robots become limited when their legs cannot make contact with the ground. In these amphibious settings, quadruped robots often struggle to generate sufficient thrust using only their legs, thereby inhibiting their functionality in such environments. While current amphibious robots are predominantly hexapods, which exhibit strong structural stability [5], their land-based mobility is often compromised. Thus, a robot with adept terrestrial locomotion and robust spatial mobility presents significant opportunities for research and application in amphibious environments.

Manuscript received: April, 30, 2023; Revised July, 24, 2023; Accepted September, 19, 2023.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the Applied Basic Research Program of Liaoning Province (2022020403-JH2/1013), the National Key Research and Development Program of China through grant 2022YFB4701900, the National Natural Science Foundation of China under Grant 61821005, the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT2023B50), the program of China Scholarship Council, and the JIANG Xinsong Innovation Fund (Grant No. E2510202).

[1]First Author, Second Author, Third Author, Fourth Author, and Eighth Author are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences and the University of Chinese Academy of Sciences, China {yaoqingfeng, menglinghan, zqf, zhaojing, wangcong2}@sia.cn

[2]Fifth Author is with the Department of Electrical Engineering and Automation, Aalto University, Finland joni.pajarinen@aalto.fi

[3]Sixth Author is with Heriot-Watt University, Edinburgh, United Kingdom xw51@hw.ac.uk

[4]Seventh Author is with University College London, London, United Kingdom alexrobotics@gmail.com

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Thrusters, providing power for aerial [6] and underwater robots [7], allow a robot to reach its desired position through vector propulsion control. Given their compact size and light weight, thrusters can be integrated with quadruped robots, enabling them to undertake tasks in amphibious environments while preserving their walking capabilities.

In underwater settings, motion control for medium-sized robots is challenging due to the nonlinear fluid dynamics and the altering dynamical effects intertwined with the robot's configuration and movements. These variables can cause significant shifts in accurately modelling the system's dynamics, thus further complicating the task of controlling the robot, especially in the presence of external perturbations.

In this paper, an adaptive propulsion framework is introduced that synchronously manages the operations of the robot's legs and thrusters, especially in dynamic fluid environments and under conditions of damage or malfunction, as illustrated in Fig. 1. The ability of our system to detect environmental changes and water flow variations relative to the robot allows the network to produce suitable adjustments and responses.

The contributions of this study are summarised as follows:

1) We introduce a unique adaptive propulsion framework that efficiently merges thrusters with the existing legged locomotion systems of quadruped robots, facilitating seamless movement in both terrestrial and aquatic settings.
2) We demonstrate the robust control algorithm that exhibits enhanced performance in speed and trajectory tracking tasks under challenging scenarios, such as actuator malfunctions and unpredictable disturbances. This algorithm equips the robot with the capability to adjust its behavior in real time.
3) We provide a thorough performance analysis of our adaptive propulsion framework, illustrating its impact and offering guidance for future research on quadruped robots in amphibious environments.

The paper is organized as follows. Section II presents the related work in the field of amphibious robots and propulsion control methods. Section III describes the proposed adaptive propulsion control method, including the integration of thrusters with the quadruped robot and the control algorithm. Section IV presents the experimental setup, the results of our training process, and the simulation results, followed by a discussion of the performance of our method compared to existing approaches. Finally, Section V concludes the paper and outlines future scope of work.

II. Related Work

Traditional control methods for legged robots emphasize modelling the robot and its environment, followed by the application of forward and inverse kinematics for position estimation and trajectory planning [8]. Reinforcement learning (RL) allows agents to learn from scratch within the environment [9], [10]. In recent years, deep reinforcement learning has been employed in quadruped robots, yielding promising results. Bellegarda et al. [11] proposed a framework for teaching quadruped robots jumping skills using RL and nonlinear trajectory optimization. Lee et al. [12] presented an RL-based controller to help legged robots navigate extreme terrains without visual input. Gangapurwala et al. [13] developed a method to plan and control the movements of quadrupeds on uneven terrains. While quadruped robots employing reinforcement learning demonstrate robust abilities in performing terrestrial locomotion tasks [2], their application in non-terrestrial environments, such as underwater exploration, is still highly desired. In aquatic settings, the locomotive stability of quadruped robots suffers due to leg capacity constraints and environmental perturbations, thereby limiting their functional capacity.

Currently, amphibious robot research is largely focused on hexapod robots owing to their inherent stability [14]. Hexapod robots, with varying structures, can achieve stable crawling in different underwater environments [15], [16]. Moreover, RL-based hexapod robots can learn swimming skills and adjust their body to attain the desired posture [17]. Nevertheless, these amphibious robots exhibit limited locomotion abilities on land [18]. By augmenting quadruped robots with the ability to traverse water surfaces, their operational range can be significantly expanded, thereby enhancing their versatility. This strategy boosts their performance while preserving their inherent dexterity and widening the scope of their potential applications.

Reinforcement learning plays a crucial role in underwater vehicles [19]. As a thruster controller for underwater vehicles [20], it has significantly enhanced underwater vehicle navigation [21]. Traditional approaches require hydrodynamic modelling of the underwater environment to achieve effective control. However, RL has the potential to accomplish tasks without establishing underwater vehicle hydrodynamics [22]. Thrusters, utilized in various underwater and spatial vehicles [23], [24], provide a significant advantage with their compact size and powerful thrust [25], [26], making them suitable for quadruped robots. Numerical research on irregular objects such as the legged robot presents challenges [27]. Certain algorithms can use the quadruped robot's historical trajectory to infer current environmental information [28]. Similar studies have affirmed the effectiveness of using historical trajectories to infer the current environmental state [29]. Hence, building on such frameworks, we employ our proposed model to estimate underwater flow conditions and the damage status of a propulsive legged robotic system.

Deep learning and reinforcement learning have been applied in the motion control of autonomous underwater vehicles (AUVs) under conditions of actuator failure and unknown disturbances [30], [31]. To adapt to rapidly changing environments, a neural network has been used to accurately predict pitch angles [32], while a model-free RL-based controller has demonstrated potential in addressing time-varying dynamics [33]. In the current study, we examined the locomotive capabilities of thrust-driven amphibious legged robots in conditions of wave interference and structural damage. Here, the primary challenge lies in the unpredictability of environmental changes. To overcome this, we introduce a teacher-student based algorithm for learning the latent environmental states. Simultaneously, we implement a prediction module to facil-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

itate the learning of environmental features and enable the prediction of current conditions. This methodology can aid thrust-driven amphibious quadruped robots in executing tasks within dynamic environments.

III. Proposed Method

A. Preliminary

In this work, Markov's decision process (MDP) [34] is expressed as $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the state transition probability, $\mathcal{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The initial state of the agent is $s_0 \in \mathcal{S}$ with a probability of $p(s_0)$. The agent selects an action $a_t \in \mathcal{A}$ according to the policy $\pi(a_t|s_t)$ at each time step t, and the environment returns a reward based on the current state and the action with $r(a_t, s_t, s_{t+1}) \in \mathcal{R}$. The agent receives a new state $s_{t+1} \in \mathcal{S}$ with a probability of $\mathcal{P} = p(s_{t+1}|s_t, a_t)$. By repeating this process, the agent obtains the trajectory $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_t, a_t, r_t, \cdots\}$. The agent updates its policy to maximize the cumulative sum of discounted rewards.

B. Propulsive Legged Robot

We introduce a vector propulsive quadruped robot structure, as shown in Fig. 2. The main part is based on a legged robot; accordingly, three joints in each leg allow the corresponding link to rotate within a specified range from the initial position. The thrusters serve as supplemental parts to supply power under the swimming condition. We attached the thrusters on the second link of the legs to acquire two degrees of freedom (DOF) and were allowed to rotate with leg movement. In addition to facilitating terrestrial locomotion, the leg joints of a quadruped robot can serve to adjust the angle of thruster propulsion in aquatic environments. As a result, the thrusters can rotate to the desired direction for vector propulsion, allowing the controller to achieve high manoeuvrability. This adaptive functionality enhances the robot's mobility across different terrain types. The lateral displacement between joints 1 and 2 is small enough to be assumed as zero; the thrust of the thruster i was applied at $r_i = (x_i, y_i, z_i)$. The controller was designed to obtain the desired propulsive force by controlling the thrust and leg angles. The dynamic effect of the thrusters can be expressed as follows:

$$F = \sum_{i=1}^{4} F_i \ , \quad T = \sum_{i=1}^{4} T_i, \tag{1}$$

where

$$T_i = r_i \times F_i \ , \quad F_i = [F_{xi}, F_{yi}, F_{zi}]^T, \tag{2}$$

where T_i is the torque vector of the thruster i, which is the result of the cross product of the thrust and positional vectors. F_i is the thrust vector of thruster i.

During its aquatic operations, the robot's thrusters generate propulsion that results in force components in various directions. This force, $F_i = [F_{xi}, F_{yi}, F_{zi}]^T$, is a vector representing thrust in different directions, which is a function of the thruster's rotation with respect to the y and z axes. Simultaneously, when a part of the robot's body (fuselage) sustains damage, it is necessary for other components of the system to adapt their output to compensate for the loss of functionality.

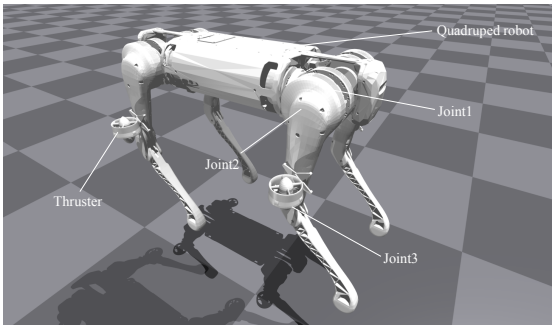

Fig. 2. Model of the vector propulsive legged robot. A thruster is designed and configured on each upper leg, endowing quadrupedal robots with the ability to move in water while maintaining their terrestrial mobility.

The dynamics of the robot can be expressed as follows:

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + w_d, \tag{3}$$

where M denotes inertia (including thruster masses), C is the Coriolis-centripetal forces (including thruster masses), and D is the damping. g is the vector of gravity and buoyancy, τ is the vector of propulsion, and w_d denotes external disturbances.

Given that the robot operates at relatively low speeds underwater, the Coriolis-centripetal forces can be disregarded to simplify calculations [35]. The damping matrix D is assumed to be diagonal due to the robot's symmetry, and any hysteresis or dead zones within the thrusters are adjusted accordingly.

C. Adaptive Propulsion Control

We have developed a control framework, referred to as adaptive propulsion control (APC), illustrated in Fig. 3. This framework is designed to enable cooperative control between legged locomotion and thruster propulsion in robotic systems. The APC takes as inputs both the latent and current states and, in turn, generates the necessary force for the thrusters and adjusts the joint angles of the robot's legs to change the direction of movement. The overall training process is organized into several stages to optimize the procedure. In the first stage, the teacher encoder and policy are trained in tandem, while the training environment undergoes random disturbances and structural damage, supplemented by privileged disturbance and fault information, denoted by p_t^{real}. The teacher encoder maps this privileged information p_t^{real} to a latent state z_t, and the policy is tasked with making decisions based on both the latent and current states.

In the subsequent stage, the student encoder and predictor are trained to produce the corresponding latent state $\hat{z}_t'$ using historical data, and to reconstruct the privileged information $\hat{p}_t'$ or $\hat{p}_t$—which includes existing disturbance and damage scenarios—from the latent state $\hat{z}_t'$ or z_t. During this stage, the parameters of the policy and the teacher encoder remain constant, while the policy employs the output of the student

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

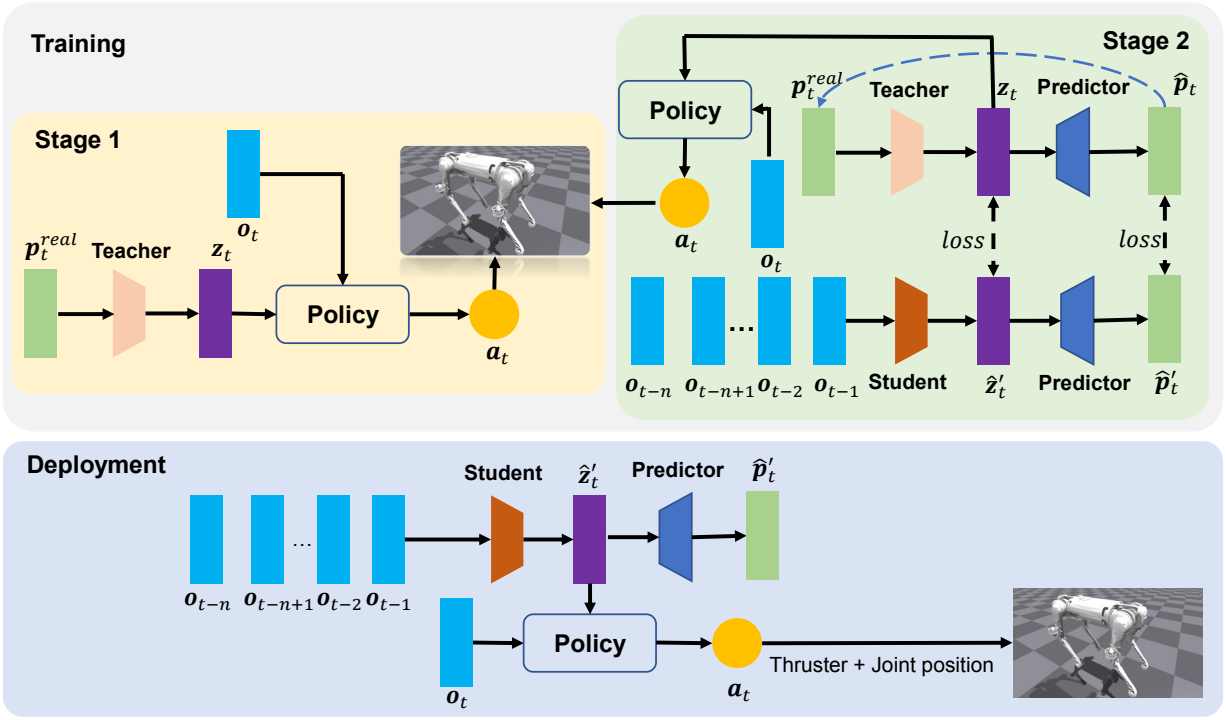

Fig. 3. Illustration of the proposed adaptive propulsion control framework. The training process is arranged in two stages. Initially, the teacher encoder learns to map environmental data into a latent state, while the policy learns to use both the *latent* and *current* states to control leg positions and generate thrust for the propeller. Subsequently, the student encoder is trained to output the *current latent state*, and the *predictor* is trained to map the latent state back to the environmental data, supervising the training of the student encoder simultaneously. Ultimately, the student encoder, policy, and predictor are applied in test scenarios, achieving both prediction and tracking control in underwater environments.

encoder, $\hat{z}'_t$, to create actions and interact with the environment. The predictor is expected to learn how to reconstruct the environmental information by minimising the ℓ_1 distance between $\hat{p}_t$ and p_t^{real}, between $\hat{p}'_t$ and p_t^{real}, and the difference between $\hat{p}_t$ and $\hat{p}'_t$. This training approach helps guide the student encoder towards convergence as follows:

$$\text{loss}_{pre} = \left\| p_t^{real} - \hat{p}_t \right\|^1 + \alpha \left\| p_t^{real} - \hat{p}'_t \right\|^1 + \beta \left\| \hat{p}_t - \hat{p}'_t \right\|^1. \tag{4}$$

The student encoder is designed to learn the environmental latent information from historical trajectories so that it closely resembles the current real latent state and can be reconstructed into environmental information by the predictor, as follows:

$$\text{loss}_{stu} = \left\| z_t - \hat{z}_t{}' \right\|^1 + \gamma \left\| \hat{p}_t - \hat{p}'_t \right\|^1. \tag{5}$$

Here, α, β, and γ serve as hyperparameters. Ultimately, the student encoder, policy, and predictor were implemented on a physical robot, enabling it to perform a wide array of tasks in unknown disturbance and damage conditions without requiring any fine-tuning. The policy takes normalized proprioceptive measurements of the robot, which consist of 57 dimensions, including base linear (3 values) and angular velocities (3 values), the estimated gravity vector (3 values), commands (4 values), historical actions (16 values), joint positions (12 values) and velocities (12 values), and the current thrust (4 values). The neural network architecture consists of a policy network, a dual-encoder system for environmental representation, and a predictor. The encoders include teacher and student encoders; this architecture has the ability to infer the characteristics of the environment [28], [29].

The teacher encoder processes specific environmental information and generates a latent state, while the student encoder estimates the current latent state based on historical data. The student representation module is implemented as a 1-D convolutional neural network (CNN) to capture temporal correlations. The remaining components are designed as multi-layer perceptron (MLP) networks, ensuring high training speed while capturing temporal relationships. Finally, the predictor reconstructs various latent states into specific environmental information regarding the current environment, such as predicting current disturbances of water flow and existing damage status. Meanwhile, the predictor assists the student encoder in completing the training.

The desired action a_t represents the required thrust for 4 thrusters and 12 leg joint positions, which are sent to the PD controller at a frequency of 50 Hz. Notably, the first two joints of each leg play a crucial role in controlling the direction of the thrusters.

D. Reward Design

The reward function is denoted as follows:

$$\begin{aligned} r_{vel} := \exp(-2(v_x - v_{x^d})^2) + \exp(-2(v_y - v_{y^d})^2) \\ + \exp(-2(v_z - v_{z^d})^2) \end{aligned} \tag{6}$$

$$r_{ang} := \exp(-2(\omega_{yaw} - \omega_{yaw^d})^2) \tag{7}$$

$$r_{orn} := -(\omega_{v_{pitch}}^2 + \omega_{v_{roll}}^2), \tag{8}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

where r_{vel} is the speed tracking reward and r_{ang} is the angle tracking reward. v_x, v_y, v_z, and ω_{yaw} denote the current linear velocity and angular velocity of the body, respectively, while v_{x^d}, v_{y^d}, v_{z^d}, and ω_{yaw^d} represent the desired linear velocity and angular velocity of the body, respectively. The angle balance reward is denoted as r_{orn}, and $\omega_{v_{pitch}}$ and $\omega_{v_{roll}}$ are the current pitch and roll rate of the body, respectively. To facilitate smoother and more efficient motion, action rate and torques are also penalized as follows:

$$r_{penalty} := -0.0001 \left\| \boldsymbol{\tau} \right\|^2 - 0.01 \left\| \mathbf{q}^* \right\|^2, \tag{9}$$

where τ represents the joint torques and q^* represents the changes in target joint positions. The power of a thruster is assumed to be proportional to the square of the thrust. All the aforementioned rewards are added together to derive the total reward, r_{total}, as follows:

$$r_{total} = 2r_{vel} + 2r_{ang} + 0.5r_{orn} + r_{penalty}. \tag{10}$$

IV. Simulation Validations

To evaluate the adaptability of the proposed adaptive propulsion control (APC) method in varying underwater environments, we conducted a series of simulation experiments. Specifically, we assessed the APC's ability to accurately track target trajectories underwater, infer water flow, and assess damage conditions. These experiments aim to demonstrate the effectiveness of the APC in handling unknown disturbances and actuator failures in underwater environments, highlighting its potential for real-world applications.

A. Fluid Dynamics

To simulate fluid dynamics in the context of an underwater robot, we considered the fluid forces and torques exerted on the robot's rigid bodies. These forces and torques are dependent on the fluid density (ρ), fluid viscosity (μ), and the geometry of the robot's structure. The fluid forces experienced by the robot are categorized into two types: (1) density-dependent forces, and (2) viscosity-dependent forces. Density-dependent forces, which are proportional to the square of the velocity, mainly contribute to lift and drag forces. Viscosity-dependent forces, linear with respect to velocity, provide additional damping effects.

For the rigid body of the robot, we compute its equivalent rectangular box, which approximates the shape of the body for fluid dynamics calculations. We then calculate the local linear (v) and angular (ω) velocities of the body in its local frame, which aligns with the equivalent rectangular box.

The density-dependent forces and torques are computed using the following formulas:

$$\begin{aligned} \text{density force } i: &\quad -\frac{1}{2}\rho s_j s_k \left| v_i \right| v_i \\ \text{density torque } i: &\quad -\frac{1}{64}\rho s_i \left(s_j^4 + s_k^4 \right) \left| \omega_i \right| \omega_i. \end{aligned} \tag{11}$$

Here, s_i, s_j, and s_k are the dimensions of the equivalent rectangular box. The index i denotes the component of the force or torque vector, and j and k are the other two indices in a cyclic permutation of $1, 2, 3$.

The viscosity-dependent forces and torques are calculated using the following formulas:

$$\begin{aligned} \text{viscosity force } i: &\quad -3\delta\pi dvi \\ \text{viscosity torque } i: &\quad -\delta\pi d^3\omega i. \end{aligned} \tag{12}$$

Here, δ is the dynamic viscosity of the fluid, and d is the average of the dimensions of the equivalent rectangular box.

B. Training Details

Initially, we trained our policies on the Isaac Gym simulator [36], which has been shown to produce effective results for quadruped robots [37]. In these experiments, we used a Unitree B1 robot [1] with 12 DOF and four T200 thrusters [2] to generate propulsive force.

The elusive environmental vector was acquired through the prediction of the latent state. To train the adaptation module, we only needed the state-action history to derive the corresponding latent states for dynamic information prediction. Both these elements can be obtained within a real environment. Random desired speed commands were generated, allowing the robot to perform tasks in different directions. Each leg joint's range of motion was $[-0.5\pi, 0.5\pi]$, and each thruster provided a maximum thrust of $50N$. If the robot's body roll or pitch exceeded 0.4 radians, the simulation was terminated prematurely without further reward, and the robot must survive as long as possible in the environment to maximize the cumulative rewards. Disturbances in different directions and thruster faults were also randomly generated during the training process. Every 4 s, there is a probability 1% that a fault occurs in each thruster, and new disturbances are generated randomly at the same time.

TABLE I
VARIATION OF PARAMETERS SAMPLED UNIFORMLY THROUGHOUT THESE RANGES DURING EACH TRAINING SESSION.

Parameters	Range	Units
X-axes velocity command	[-1, 1]	m/s
Y-axes velocity command	[-0.5, 0.5]	m/s
Z-axes velocity command	[-0.5, 0.5]	m/s
Angular command	$[-\pi, \pi]$	rad
Disturbing force	[-50, 50]	N
Joint positions noise	[-0.1, 0.1]	rad
Joint velocities noise	[-1.5, 1.5]	rad/s
Linear velocity noise	[-0.02, 0.02]	m/s
Angular velocity noise	[-0.05, 0.05]	rad/s

To enhance the model's generalization capabilities, we randomized several components, including parameters and observation noise. The detailed parameters are provided in TABLE I.

The student encoder's input was the robot state and actions from the past 50 timesteps. The student encoder employed two 1-D convolutional layers across the time dimension to capture temporal correlations. The output channel number, kernel size, and stride of each layer were [32, 3, 2] and [32, 5, 1], respectively. The flattened CNN output was then used

[1] https://shop.unitree.com/products/unitree-b1

[2] https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

to predict $\hat{z}_t'$ with a linear layer. The values of α, β and γ were set to 0.2. The teacher encoder and predictor used a two-layer neural network, with each layer comprising 32 neuronal nodes. We employed the Adam optimizer [38] to minimize the mean absolute error (MAE) loss. We adopted the proximal policy optimization (PPO) [39] method for the policy. The experiment had 4096 robots training concurrently in parallel, employing a two-layer neural network with a learning rate of 5e-4. Each network layer contained 32 neuronal nodes and utilized the Adam optimizer. The student and predictor modules were trained on 500 robots for 500 episodes.

C. Simulation Results

We achieved 1500 policy updates utilizing a single NVIDIA RTX 3080 GPU. The velocity tracking performance achieved by the APC is depicted in Fig. 4. These results demonstrate that our method, even amidst disturbances, enables the thrusters and legs to collaboratively track the desired speed. The results also suggest that our proposed method is capable of resisting water disturbances.

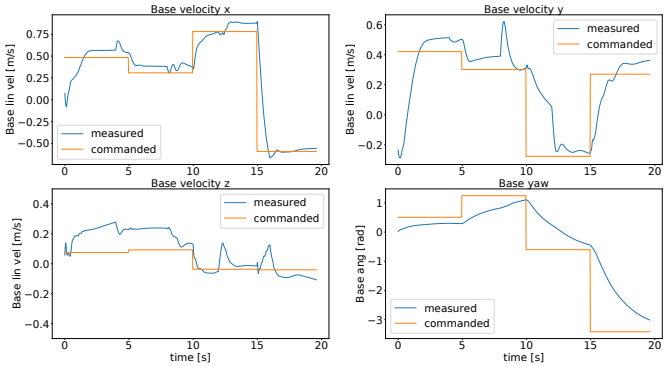

Fig. 4. Result of speed tracking effect of APC. The APC can accurately track the linear velocity in the x-axis, y-axis, and body yaw.

We evaluated the APC's predictive capabilities and its effectiveness in adjusting actions when confronted with random disturbances and faults. The prediction results are displayed in Fig. 5. As shown, the trend of disturbances in different directions can be predicted via the latent state. Besides disturbances, actuation failures also impact tracking performance. Thus, the network must use the latent state to predict current fault conditions. We evaluated scenarios involving sudden faults during motion and visualized the prediction output for actuator fault tasks. Simulating a propeller failure during motion shows that the network can predict the damage based on historical trajectory data. Simultaneously, the agent can adjust its output to perform tracking tasks.

We evaluated various trajectory tracking tasks. For testing the robot's dexterity, we used a line and a cycle trajectory, as illustrated in Fig. 7. Sudden disturbances occur during operation. Here, it was observed that the APC could swiftly adjust its actions and accomplish trajectory tracking upon encountering disturbances, whereas the PPO method deviated from the trajectory due to disturbances. The results reveal that the legs and thrusters can collaborate to achieve smoother performance in the face of disturbances, which is a challenge

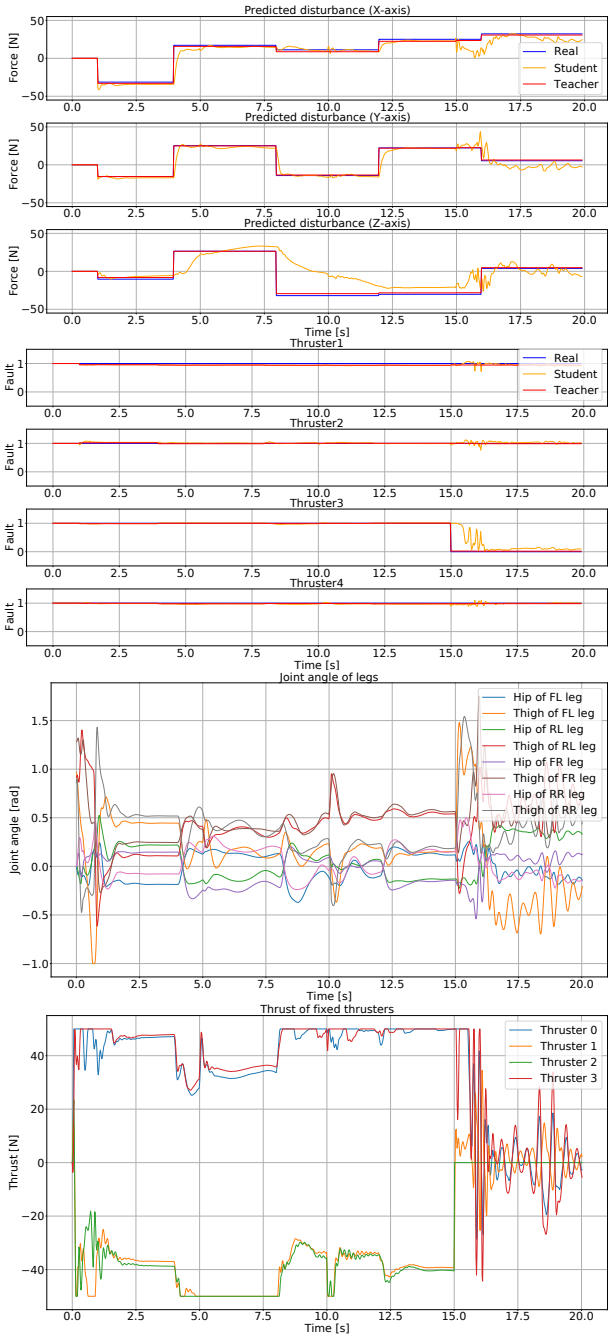

Fig. 5. Predicted disturbance and faults, along with the actions of the thrusters and leg joint positions. When disturbances and failures arise, the APC can adaptively adjust its actions to counteract the effects, thus providing predictive ability to various perturbations.

for the PPO method. We demonstrate the process of the robot following a circular trajectory in Fig. 6, showing that the robot can flexibly turn to complete the circular path tracking.

We evaluated the effect of the latent dimension on the results by training with latent state dimensions of 16, 32, 64, and compared the results with PPO and conducted ablation studies, as presented in Table II. The results present the average reward values under different disturbances. We calculated the mean of the results for each experiment after 50 runs to illustrate this effect. Concurrently, we tested the effect of different sizes of perturbations on the results. We found that under conditions

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

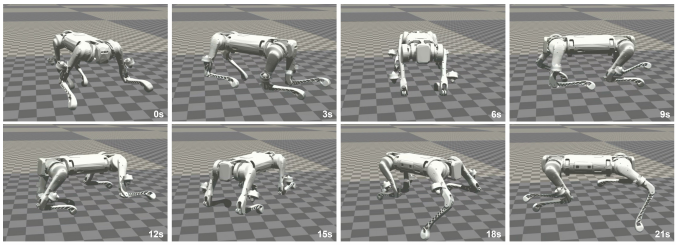

Fig. 6. Snapshots of the APC trajectory tracking performance, showcasing the robot's ability to accurately follow a circular path by executing turns.

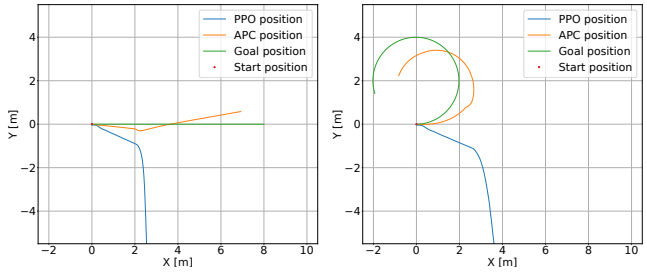

Fig. 7. Tracking effects of velocity tracking based on different methods. Under the influence of perturbations, the PPO deviates from the intended trajectory. However, due to its predictive capacity and dynamic action adjustments, the APC can still accomplish trajectory tracking amidst disturbances.

of perturbations and faults, the performance of PPO was relatively poor, whereas the proposed method demonstrated more reasonable results. This indicates that by relying solely on the domain randomization technique, PPO fails to achieve optimal results, and the APC method exhibited increased robustness for varying magnitudes of perturbations. In addition, a latent state dimension of 32 offers the highest overall performance; this indicates that increasing or decreasing the dimensionality of the latent state does not guarantee improved performance and robustness. We further conducted a comparative analysis of the disturbance prediction capabilities of APC and the model without reconstruction (w/o Re), both set with 32 latent state dimensions. Err(A) denotes the disturbance prediction error for APC and Err(w) represents the error for the model without the reconstruction of environmental features, the results reflect the average estimation error of disturbances in various directions, expressed in Newtons (N). The results highlight that, while APC ensures performance, it exhibits superior capabilities in environmental prediction.

To understand the estimated perturbed latent states, we evaluated the clustering of the latent state vectors under distinct scenarios. We sampled the robot's latent state under various conditions, including different thruster malfunctions and disturbances from multiple directions. In Fig. 8, we used t-SNE to visualize the estimated latent states under different disturbances and faults. The results reveal that the latent states in different environments frequently occupy separate regions and exhibit unique characteristics.

V. Conclusion and Future Work

In this study, we successfully developed an adaptive propulsion control (APC) framework that allows for the integration of thrusters with quadruped robots. This integration enables the robot to move seamlessly between terrestrial and aquatic environments. The key contributions of our work are coordinated

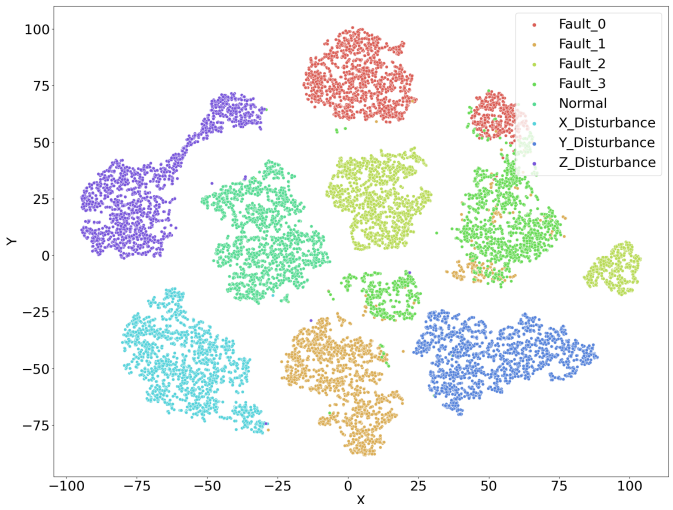

Fig. 8. The t-SNE result of the estimated latent state for different disturbances or various damage situations. The APC generates distinctive latent states based on the unique characteristics of various types of disturbances and failures, enhancing the controller's versatility and adaptability in dynamic environments.

TABLE II
ANALYSIS OF LATENT STATES ACROSS DIMENSIONS SHOWS THAT LOW OR HIGH DIMENSIONS DEGRADE THE MODEL PERFORMANCE AND APC DEMONSTRATES SUPERIOR ENVIRONMENTAL PREDICTION CAPABILITIES WHILE MAINTAINING ITS PERFORMANCE.

Force	16	32	64	PPO	w/o Re	Err(A)	Err(w)
0N	9.22	9.51	9.56	6.50	9.41	7.39	5.68
30N	9.25	9.43	9.48	6.29	9.35	6.88	4.45
50N	8.95	9.24	9.21	5.85	9.17	6.93	4.36
80N	7.98	8.28	8.11	3.92	8.15	12.12	8.90

control of the legs and propellers, the adaptive neural network to predict and compensate for disturbances and actuation failures, and simulation experiments that demonstrated the ability to effectively perform speed and trajectory tracking tasks, even under challenging conditions.

However, while our simulation experiments were successful, there are many more possibilities to explore. We are looking towards conducting real-world experiments, which will require significantly more resources, facilities, and hardware support. Future studies could consider incorporating additional sensing and control modalities to improve performance in complex environments. We are also interested in performing a deeper analysis of the transition mode between landing and swimming.

In conclusion, our proposed method shows substantial potential for the hardware augmentation of quadruped robots, creating new applications in amphibious environments, and opening new areas of research in the field of robotics.

References

[1] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[2] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[3] Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8352–8358. IEEE, 2021.

[4] Hendrik Kolvenbach, Manuel Breitenstein, Christian Gehring, and Marco Hutter. Scalability analysis of legged robots for space exploration. In *Unlocking imagination, fostering innovation and strengthening security: 68th International Astronautical Congress (IAC 2017)*, volume 16, pages 10399–10413. Curran, 2018.

[5] Bin Zhong, Shiwu Zhang, Min Xu, Youcheng Zhou, Tao Fang, and Weihua Li. On a cpg-based hexapod robot: Amphihex-ii with variable stiffness legs. *IEEE/ASME Transactions on Mechatronics*, 23(2):542–551, 2018.

[6] Marco Tognon and Antonio Franchi. Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control. *IEEE Robotics and Automation Letters*, 3(3):2277–2282, 2018.

[7] Jeongae Bak, Yecheol Moon, Jongwon Kim, Santhakumar Mohan, TaeWon Seo, and Sangrok Jin. Hovering control of an underwater robot with tilting thrusters using the decomposition and compensation method based on a redundant actuation model. *Robotics and Autonomous Systems*, 150:103995, 2022.

[8] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[10] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[11] Guillaume Bellegarda and Quan Nguyen. Robust quadruped jumping via deep reinforcement learning. *arXiv preprint arXiv:2011.07089*, 2020.

[12] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020.

[13] Siddhant Gangapurwala, Mathieu Geisert, Romeo Orsolino, Maurice Fallon, and Ioannis Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 2022.

[14] Alberto Perez Nunez, Matko Orsag, and Daniel M Lofaro. Design, implementation, and control of the underwater legged robot aquashoko for low-signature underwater exploration. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 228–234. IEEE, 2018.

[15] Hyungwon Shim, Seong-yeol Yoo, Hangoo Kang, and Bong-Huan Jun. Development of arm and leg for seabed walking robot crabster200. *Ocean Engineering*, 116:55–67, 2016.

[16] Heejoong Kim and Jihong Lee. Design, swimming motion planning and implementation of a legged underwater robot (caleb10: D. beebot) by biomimetic approach. *Ocean engineering*, 130:310–327, 2017.

[17] David Meger, Juan Camilo Gamboa Higuera, Anqi Xu, Philippe Giguere, and Gregory Dudek. Learning legged swimming gaits from experience. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2332–2338. IEEE, 2015.

[18] Ioannis Davliakos, Ioannis Roditis, Klajd Lika, Ch-M Breki, and Evangelos Papadopoulos. Design, development, and control of a tough electrohydraulic hexapod robot for subsea operations. *Advanced Robotics*, 32(9):477–499, 2018.

[19] Ignacio Carlucho, Mariano De Paula, Sen Wang, Yvan Petillot, and Gerardo G Acosta. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 107:71–86, 2018.

[20] Tao Liu, Yuli Hu, and Hui Xu. Deep reinforcement learning for vectored thruster autonomous underwater vehicle control. *Complexity*, 2021, 2021.

[21] Ignacio Carlucho, Mariano De Paula, Sen Wang, Bruno V Menna, Yvan R Petillot, and Gerardo G Acosta. Auv position tracking control using end-to-end deep reinforcement learning. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–8. IEEE, 2018.

[22] Gang Chen, Yuwang Lu, Xin Yang, and Huosheng Hu. Reinforcement learning control for the swimming motions of a beaver-like, single-legged robot based on biological inspiration. *Robotics and Autonomous Systems*, 154:104116, 2022.

[23] Xichuan Lin and Shuxiang Guo. Development of a spherical underwater robot equipped with multiple vectored water-jet-based thrusters. *Journal of Intelligent & Robotic Systems*, 67(3):307–321, 2012.

[24] Xin-hui Zheng, Cong Wang, Xue-jiao Yang, Qi-yan Tian, Qi-feng Zhang, and Bao-qi Zhai. A novel thrust allocation method for underwater robots. In *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 271–276. IEEE, 2022.

[25] Kyunam Kim, Patrick Spieler, Elena-Sorina Lupu, Alireza Ramezani, and Soon-Jo Chung. A bipedal walking robot that can fly, slackline, and skateboard. *Science Robotics*, 6(59):eabf8136, 2021.

[26] Hosameldin Awadalla Omer Mohamed, Gabriele Nava, Giuseppe L'Erario, Silvio Traversaro, Fabio Bergonti, Luca Fiorio, Punith Reddy Vanteddu, Francesco Braghin, and Daniele Pucci. Momentum-based extended kalman filter for thrust estimation on flying multibody robots. *IEEE Robotics and Automation Letters*, 7(1):526–533, 2021.

[27] Wenlong Tian, Baowei Song, and Hao Ding. Numerical research on the influence of surface waves on the hydrodynamic performance of an auv. *Ocean Engineering*, 183:40–56, 2019.

[28] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[29] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pages 1722–1732. PMLR, 2023.

[30] Fei Huang, Jian Xu, Liangang Yin, Di Wu, Yunfei Cui, Zheping Yan, and Tao Chen. A general motion control architecture for an autonomous underwater vehicle with actuator faults and unknown disturbances through deep reinforcement learning. *Ocean Engineering*, 263:112424, 2022.

[31] Xiaofei Chang, Lulu Rong, Kang Chen, and Wenxing Fu. Lstm-based output-constrained adaptive fault-tolerant control for fixed-wing uav with high dynamic disturbances and actuator faults. *Mathematical Problems in Engineering*, 2021:1–18, 2021.

[32] Shanshan Song, Jun Liu, Jiani Guo, Jun Wang, Yanxin Xie, and Jun-Hong Cui. Neural-network-based auv navigation for fast-changing environments. *IEEE Internet of Things Journal*, 7(10):9773–9783, 2020.

[33] Peng Jiang, Shiji Song, and Gao Huang. Attention-based meta-reinforcement learning for tracking control of auv with time-varying dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6388–6401, 2021.

[34] Martijn van Otterlo and Marco Wiering. Reinforcement learning and markov decision processes. In *Reinforcement learning*, pages 3–42. Springer, 2012.

[35] Thor I Fossen. Marine control systems–guidance. navigation, and control of ships, rigs and underwater vehicles. *Marine Cybernetics, Trondheim, Norway, Org. Number NO 985 195 005 MVA, www. marinecybernetics. com, ISBN: 82 92356 00 2*, 2002.

[36] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[37] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.