IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Reinforcement Learning for Underwater Spatiotemporal Path Planning, with Application to an Autonomous Marine Current Turbine

Arezoo Hasankhani[1], Yufei Tang[1], and James VanZwieten[2]

***Abstract*— This paper presents a reinforcement learning (RL) framework applied for an autonomous underwater vehicle (AUV) path planning, focusing on a specific type of energy-harvesting AUV, entitled marine current turbine (MCT). The proposed RL-based approach improves a classical path planning to adopt with an underwater environment prone to spatiotemporal uncertainties. The path planning problem is formulated to achieve the goal of maximizing the harnessed energy from the MCT subject to the agent dynamics and the spatiotemporal environment constraints. Three RL algorithms, including Q-learning, deep Q-network (DQN), and proximal policy optimization (PPO), are nominated to deal with the path planning over both discrete gridded and continuous underwater environments modeling. The experimental results demonstrate the efficiency of the RL-based approaches in seeking the optimal path in the underwater environment, where further discussion is presented to generalize the proposed approach to other energy-harvesting autonomous vehicles operating in the spatiotemporally varying environment, such as airborne wind turbines.**

I. INTRODUCTION

The goal of path planning is to generate a feasible and valid path for an autonomous agent to perform a specific mission and achieve an ultimate objective. In the field of autonomous agents path planning, an autonomous underwater vehicle (AUV) has gained increasing attention due to its complexity, and lack of human accessibility [1]. The AUV systems have been primarily used for searching and investigation in the underwater environment, implying minimized power consumption and travel time [2]. This paper focuses on a recently developed application of AUVs for harnessing renewable energy from underwater environment [3].

The path planning problem becomes more complicated for the autonomous agent vehicles operating in a spatiotemporally varying environment [4], [5], [2], [1], especially for the emerging field of energy-harvesting autonomous vehicles, such as an airborne energy system [6], [7], ocean kite [8], and marine current turbine (MCT) [9], [10]. In these applications, the classical path planning algorithms, i.e., graph searching techniques designed originally for the autonomous agents operating in a static gridded environment [11], [12], are not adequate in a spatiotemporally varying environment. Hence, it is intuitive to develop an approach to fit a particular task of seeking an optimal path in an environment with enhanced space-time uncertainties.

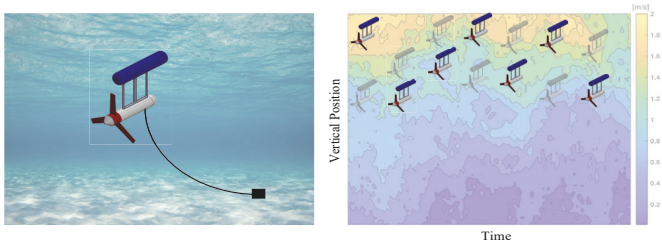

Fig. 1: (Left) Autonomous MCT agent that has been investigated in this work; (Right) Overall schematic of the path planning problem in a spatiotemporally varying underwater environment, where the contour plot visualizes the ocean current speed.

The first attempts to address the path planning problem for autonomous renewable energy agents have been devoted to the airborne energy system, where an extremum seeking approach [7] and model predictive control (MPC) method [13] have been proposed to find a sequence of optimal waypoints. Similar problem has been then formulated in the context of the energy-harvesting ocean kite [8] addressed through the MPC approach. Finally, in the most recent application of MCT, both MPC and reinforcement learning (RL) have been used to find the optimal path [14]. The predictive approach builds upon the forecasting of the spatiotemporally varying environment, limiting the path planning to the forecasting accuracy. It is then favorable to take a look at learning-based algorithms that directly learn from real recorded data in an uncertain environment.

Among many learning-based methods, RL has been introduced as a promising nominee to cope with the planning problem [15]. One of the most prevalent challenges existing in the RL frameworks is to craft an appropriate reward function to precisely describe the ultimate objective of the problem at hand. The reward function usually serves as an objective function in the conventional optimization problem, and a large body of research in path planning has been devoted to defining and solving an appropriate objective function [16], [17]. The primary task in this paper is to extend the ideas from objective functions in path planning optimization problems to develop a favorable reward function for energy harnessing. Although there exists research on using the RL for the path planning of AUVs with the major goal of collision avoidance [18], [19], [20], [21], [22], RL-

This work was supported in part by the National Science Foundation under Grant No. CMMI-2145571 and the U.S. Department of Energy under Grant No. DE-EE0008955.

[1]Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA. {ahasankhani2019, tangy}@fau.edu.

[2]Department of Ocean and Mechanical Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA. jvanzwi@fau.edu.

based path planning for power maximization is still lacking.

The RL can leverage either discrete action space or continuous action space to handle the path planning. The action space here interprets the optimal path characterized in the underwater environment, where there exist two avenues to select the optimal path: (i) a set of predefined waypoints in a discrete gridded environment; and (2) any point in a continuous environment. The first avenue can be pursued through the RL approaches dealing with the discrete space, such as Q-learning [23], [24] and deep Q-network (DQN) [14]. Additionally, policy gradient-based methods like deep deterministic policy gradient [25] and proximal policy optimization (PPO) [20] can be employed to address the latter. The room is then left for a comparative analysis on the theoretical aspect and experimental results for the RL algorithms over discrete and continuous environments, considering this paper introduces a new application on path planning of an autonomous MCT.

The purpose of this paper is to fill the gap in the development of an RL-based approach for path planning in an environment with enhanced space-time uncertainties. The case study focuses specifically on the autonomous MCT (as depicted in Fig. 1(a)), with the ultimate goal of placing the agent at the operating depth that maximizes the total accumulated power production in a given time range. This vertical path planning problem of the agent operating in the spatiotemporally varying underwater environment is visualized in Fig. 1(b). Note that this paper focuses exclusively on the path planning for the autonomous MCT, relying on the authors' prior work [26] on developing the path tracking controller for the autonomous MCT that ensures navigating the agent through the optimal path. Moreover, this paper furthers the authors' previous work [24] on the spatiotemporal path planning for the MCT through exploring and evaluating different RL candidates, designing proper reward functions for each candidate, and eventually discussing the extension to other energy-harvesting autonomous vehicles. The previous study [24] has been mainly devoted to (i) modeling of power equation and justifying the linear vertical movement with a nonlinear model, and (ii) designing a Q-learning algorithm to solve the spatiotemporal path planning.

Contribution: The main contribution of this paper is to present an RL-based framework for path planning in the spatiotemporally varying underwater environment while performing comparative analysis on different RL algorithms. Our intention is to open up a new perspective on performance evaluation of RL approaches (i.e., Q-learning, DQN, and PPO) enabled for path planning over the discrete gridded and continuous environments. In a proposed case study of the autonomous MCT, path planning is employed to maximize the harnessed power from the agent, which is finally discussed to be extended to a more general application of the energy-harvesting autonomous vehicles operating in the uncertain environment.

II. AUTONOMOUS MCT AND UNDERWATER ENVIRONMENT MODELING

A. Marine Current Turbine Modeling

This paper investigates a specific case study of a 700 kW autonomous MCT [9], which is tethered to the seafloor and can be controlled and moved in a vertical direction to harness the power from the underwater environment. The MCT agent, consisting of the main body, buoyancy tank, rotor, and mooring cable, with a detailed dynamic model, equations of motion, and numerical simulation are given in [9], [26]. The MCT agent is interpreted with 14 states ($x \in \mathbb{R}^{14}$) and 3 control inputs ($u \in \mathbb{R}^3$). Given the state vector $x = [u\ v\ w\ p_\mathrm{b}\ p_\mathrm{r}\ q\ r\ x\ y\ z\ \phi_\mathrm{b}\ \phi_\mathrm{r}\ \theta\ \psi]^\top$, with $[x\ y\ z\ u\ v\ w]^\top$ being the linear position and velocity of MCT body, $[\phi_\mathrm{b}\ \theta\ \psi\ p_\mathrm{b}\ q\ r]^\top$ being the attitude and velocity of MCT body, and $[\phi_\mathrm{r}\ p_\mathrm{r}]^\top$ being the attitude and velocity of the MCT rotor. Also, consider the control inputs $u = [B_f\ B_\mathrm{a}\ \tau_\mathrm{em}]^\top$, with B_f and B_a being the forward and aft buoyancy tank fill fractions, and τ_em being the electromechanical torque. The vertical movement of the MCT is primarily controlled with B_f and B_a. Note that the flight controller of the investigated MCT is precisely described in our previous study [26], where we demonstrate that defining constraints on the allowable operating depth and rate of changes in the operating depths is enough to ensure a feasible path for the MCT. The main objective of the flight controller is to track the optimal path commanded by the path planning algorithm with minimum error and find the optimal control inputs.

For the path planning, let us concentrate on the vertical movement of the MCT (characterized with the vertical position z) and its interaction with the fill fractions as leading actuators in the MCT movement, as well as underwater current velocity v_e. To form a linear interaction, the nonlinear dynamic model is approximated with the following linear equation (see [24] for details and justification between nonlinear and linear models):

$$\Delta B_{(.)} = \alpha_1 \Delta v_\mathrm{e} + \alpha_2 \Delta z \tag{1}$$

where this equation interprets that vertical movement z is a function of current velocity v_e and fill fraction $B_{(.)}$, where the equilibrium point has a current velocity of 1.6 m/s and a vertical position of 50 m. α_1 and α_2 are the constant coefficients approximated by the nonlinear model. Moreover, equal values are assumed for fill fractions $B_{(.)}$. Using a similar approach, the harnessed power from the MCT is approximated with three terms of produced power P_P, consumed power to hold the vertical position due to changes in the current velocity P_HD, and consumed power to change the vertical position P_CD, as shown in the following equations (see [24] for details):

$$P_\mathrm{net} = P_\mathrm{P} - P_\mathrm{HD} - P_\mathrm{CD} \tag{2a}$$

$$P_\mathrm{P} = \mathrm{clip}(\frac{1}{2}\rho A v_\mathrm{e}^3 c_\mathrm{p}, P_\mathrm{r}) \tag{2b}$$

$$P_\mathrm{HD} = \begin{cases} 0, & \Delta v_\mathrm{e} < 0 \\ \frac{\alpha_1}{T_\mathrm{s}} \Delta v_\mathrm{e}, & \Delta v_\mathrm{e} \geq 0 \end{cases} \tag{2c}$$

$$P_{\mathrm{CD}} = \begin{cases} 0, & \Delta z > 0 \\ \frac{\alpha_2}{T_{\mathrm{s}}} \Delta z, & \Delta z \leq 0 \end{cases} \tag{2d}$$

where ρ denotes the water density, A denotes the rotor area, c_{p} denotes the power coefficient, P_{r} denotes the rated power, and T_{s} is the sampling time.

B. Underwater Environment Modeling

The path planning problem can be solved in an environment characterized by either discrete gridded space or continuous space. The gridded space entails prior knowledge on the underwater current velocity v_{e} at specific points of the discrete environment denoted by $z_{\mathrm{e}}^{\mathrm{d}}$, which is durable using the field recorded data by an acoustic Doppler current profiler (ADCP). The gridded spatiotemporally varying environment is limited to finite size of $n \times h$, given n discrete spatial positions and h as a horizon in the temporal domain. Unlike the discrete environment, the continuous space is constructed with an assumption of a fully observable environment and complete knowledge of the current velocity at any spatial points denoted by z_{e}. It is worthwhile to mention that the continuous space is illustrated with an infinite spatial point but in a finite horizon h similar to the discrete space.

III. REINFORCEMENT LEARNING APPROACH FOR PATH PLANNING

A. Preliminaries on Reinforcement Learning

RL is an approach to deal with sequential decision-making problems through trial and error and making final decision based on the experience acquired by performing actions in an uncertain environment and gained rewards [27]. RL is a promising candidate for the path planning in the spatiotemporally varying underwater environment. The agent observes the environment illustrated by states $s \in S$, with S being the set of states, accomplish an action $a \in A$ from an action set A following a policy mapping the states to the actions $\pi(a|s) : S \to A$, thus the environment transitions to a new state $s' \in S$. This state transition gains scalar feedback entitled a reward r. The agent's goal is to learn the optimal policy π^* to maximize the cumulative reward over a horizon T, $\mathscr{R} = \sum_{k=0}^{T} \gamma^k R_{k+1}$, with $\gamma \in [0,1]$ being the discount factor, demonstrating the priority of immediate rewards over later ones. To quantify the state value, two functions are introduced as: (i) state-value function V, estimating the expected return when starting at s and following π; and (ii) action-value function Q, calculating the expected return starting at s, following π, and taking action a, namely:

$$V^{\pi}(s) = \mathbb{E}[\mathscr{R}|s,\pi] \tag{3}$$

$$Q^{\pi}(s,a) = \mathbb{E}[\mathscr{R}|s,a,\pi] \tag{4}$$

B. Recast of MCT Path Planning Into RL Framework

In the path planning problem for MCT, the ultimate objective of RL is to endow MCT with the ability to learn how to find the optimal path that maximizes the harnessed power from the agent. The first task is to define the state set, action set, and reward function for the problem at hand. The state set is illustrated by the MCT position and underwater environment current velocity at t, i.e., $S = \{z(t), v_{\mathrm{e}}(t)\}$. The action space is defined as a vector of feasible vertical positions for the MCT, where an action taken at t should enforce the MCT to reach the specified position at $t+1$. The action space is formulated by $A = \{z_{\mathrm{e}}(t)\}$, given that $z_{\mathrm{e}}(t) \triangleq z_{\mathrm{e}}^{\mathrm{d}}$ for the discrete gridded environment.

It is favorable to tune shaping reward functions with demonstrations of the ultimate objective of power maximization, which is separately formulated for each candidate RL algorithm to achieve the best performance. Since each RL algorithm follows a specific approach to find the optimal policy, it is predictable that the same reward function may not yield the best results for all methods. The experimental results for other applications also show the RL algorithm's performance highly depends on the reward function. Note that different reward functions are tested, and the best reward function for each algorithm is presented here.

Three RL algorithms, including two approaches over the discrete environment and one for the continuous environment, are nominated to solve the MCT path planning problem. All algorithms are initially trained offline using the field-recorded data from the Gulf Stream, which is then applied in the online path planning.

Q-learning Algorithm: The Q-learning algorithm is assigned as the baseline, dealing with the problems defined over the gridded environment with a set of discrete actions. This algorithm employs a Q-table to store $Q(s,a)$ for all feasible states and actions, where the highest Q-value determines the optimal action taken at each state. The reward function to fulfill the power maximization is defined as follows:

$$R^{\mathrm{Q}} = \begin{cases} P_{\mathrm{net}} - P_{\mathrm{net}}^{\mathrm{b}}, & P_{\mathrm{net}} - P_{\mathrm{net}}^{\mathrm{b}} > \delta_1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where an action is rewarded while the power change is greater than a constant threshold of δ_1, and $p_{\mathrm{net}}^{\mathrm{b}}$ denotes a base net power parameter while keeping the current vertical position (i.e., no vertical movement).

To balance the exploration and exploitation, an ε-greedy approach is used [14], namely:

$$a = \begin{cases} \arg\max_a Q^{\pi}(s,a), & 1-\varepsilon \\ \text{random } a, & \varepsilon \end{cases} \tag{6}$$

where $\varepsilon = \varepsilon_{\min} + (\varepsilon_{\max} - \varepsilon_{\min}) e^{-d \cdot e}$, with d being the decay factor, and e being the episode. The training of the Q-learning application for path planning is presented in Algorithm 1.

Deep Q-Network (DQN) Algorithm: The DQN approximates Q function with a neural network to deal with a large size state space and action space. This algorithm utilizes two neural networks with the same structure but different weights, where the weight of "target network" $Q(s,a|\theta'^{\mathrm{DQN}})$ is updated using the "Q-network" $Q(s,a|\theta^{\mathrm{DQN}})$. The DQN learns the optimal Q^* function through the optimal policy acquired by minimizing the following loss function:

$$\mathscr{L}(\theta^{\mathrm{DQN}}) = [Q(s(t),a(t);\theta'^{\mathrm{DQN}}) - Q(s(t),a(t);\theta^{\mathrm{DQN}})]^2 \tag{7}$$

Algorithm 1 Q-learning for path planning

Input: field-recorded current velocity, discrete spatial positions, and Q-learning parameters;
Output: optimal Q-table;
for each episode **do**
  Sample an initial state $s \leftarrow \{z, v_e\}$;
  for each step of episode **do**
    Accomplish $a \leftarrow z_e^d$ using ε-greedy policy (6);
    Gain R^Q by (5), and update Q and s;
  end for
end for

with

$$Q(s(t),a(t);\theta'^{DQN}) \triangleq R(t+1)+\gamma \max_a Q(s(t+1),a;\theta'^{DQN}) \quad (8)$$

The reward function for the DQN algorithm is defined by two positive constant values ζ_1 and ζ_2 to avoid a high increase in the cumulative reward; the reward is given to the agent if it follows the increased velocity, as well as the increased power. This shaping of the reward function accelerates the training of the DQN by identifying a trend for the velocity increase and accordingly the power increase. The reward function is defined as follows:

$$R^{DQN} = \varpi_1 R_P^{DQN} + \varpi_2 R_v^{DQN} \quad (9)$$

with

$$R_P^{DQN} = \begin{cases} \zeta_1, & P_{net} - P_{net}^b > \delta_1 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$R_v^{DQN} = \begin{cases} \zeta_2, & v_e - v_e^b > \delta_2 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where the reward function consists of two terms due to the reward for power R_P^{DQN}, as well as velocity R_v^{DQN} with ϖ_1 and ϖ_2 showing the constant coefficients; v_e^b similar to p_{net}^b defined in (5) denotes a base environment velocity parameter while holding the current vertical position.

The action is selected based on the same ε-greedy policy defined in (6). The algorithm of DQN for offline training is illustrated in Algorithm 2, where $\mathscr{D}$ denotes the experience replay memory.

Proximal Policy Optimization (PPO) Algorithm: To cope with the continuous states and actions, the PPO algorithm [28] is adopted in this paper. Let define the advantage function $\mathscr{A}$ as the difference between state-value and action-value functions $\mathscr{A}(s,a) = Q(s,a) - V(s)$. To build an estimate from the advantage function at t denoted as $\widehat{\mathscr{A}(t)}$, a critic network is used to approximate the value function $\widehat{V}$, and the advantage function estimation $\widehat{\mathscr{A}(t)}$ is defined by:

$$\widehat{\mathscr{A}(t)} = \delta(t) + (\gamma\lambda)\delta(t+1) + \ldots + (\gamma\lambda)^{T-t+1}\delta(T-1) \quad (12)$$

with

$$\delta(t) = R(t) + \gamma\widehat{V}(s(t+1)) - \widehat{V}(s(t)) \quad (13)$$

where T denotes the time horizon, γ denotes the discount factor, $0 \le \lambda \le 1$ denotes a parameter to bias the variance

Algorithm 2 DQN for path planning

Input: field-recorded current velocity, discrete spatial positions, and DQN parameters;
Output: optimal DQN;
for each episode **do**
  Sample an initial state $s \leftarrow \{z, v_e\}$;
  for each step of episode **do**
    Accomplish $a \leftarrow z_e^d$ using ε-greedy policy (6);
    Gain R^{DQN} by (9), and update s;
    Store transition (s,a,r,s') in $\mathscr{D}$;
    Sample random mini-batch from $\mathscr{D}$;
    Calculate target Q-value by (8);
    Perform a gradient descent step on loss in (7);
    Update target network and update $\theta'^{DQN} \leftarrow \theta^{DQN}$ every c_1 steps;
  end for
end for

trade-off. The ultimate goal of the PPO is to maximize a "surrogate objective function", formulated as follows:

$$L^{CLIP}(\theta^{PPO}) = \widehat{\mathbb{E}}_t[\min(r(t;\theta^{PPO})\widehat{\mathscr{A}(t)}, \text{clip}(r(t;\theta^{PPO}), 1-\vartheta, 1+\vartheta)\widehat{\mathscr{A}(t)})] \quad (14)$$

with $r(t;\theta^{PPO}) = \frac{\pi(a,s;\theta^{PPO})}{\pi(a,s;\theta'^{PPO})}$ being the probability ratio with $\pi(a,s;\theta'^{PPO})$ representing the old policy, which is clipped to stay within a constant range of $[1-\vartheta, 1+\vartheta]$. The reward function is defined according to the continuous nature of the action and state spaces:

$$R^{PPO} = \varsigma_1 R_P^{PPO} + \varsigma_2 R_v^{PPO} \quad (15)$$

with

$$R_P^{PPO} = \text{clip}(\frac{P_{net} - P_{net}^{des}}{P_{net}^{des}}, -1, +1) \quad (16)$$

$$R_v^{PPO} = \text{clip}(\frac{v_e - v_e^{des}}{v_e^{des}}, -1, +1) \quad (17)$$

where the reward function includes two terms corresponding to the velocity and power with ς_1 and ς_2 denoting the constant coefficients; v_e^{des} and P_{net}^{des} denote large values as desired velocity and power.

The PPO algorithm for the path planning is outlined in Algorithm 3.

IV. Experimental Results

A. Simulation Setup

The simulations are carried out for a sample autonomous MCT agent, where the whole design parameters are given in [9]. The main parameters corresponding to the linear movement relation are $\alpha_1 = 0.65$ s/m, $\alpha_2 = -0.0026$ 1/m, $\rho = 1024$ kg/m^3, $A = 100\pi$ m^2, $C_p = 0.415$, and $T_s =$ 1 hour. For the RL algorithms, the primary parameters in the simulation include $\delta_1 = 1$ kW, $\varepsilon_{min} = 0.01$, $\varepsilon_{max} = 1$, $d = 0.01$, $e = 3000$, $\varpi_1 = 1$, $\varpi_2 = 0.5$, $\zeta_1 = 1$, $\zeta_2 = 1$, $\delta_1 =$ 1 kW, $\delta_2 = 0.001$ m/s, $\varsigma_1 = 0.8$, $\varsigma_2 = 0.2$, $P^{des} = 700$ kW, $v_e^{des} = 2$ m/s. The discount factors for Q-learning, DQN, and PPO algorithms are $\gamma^Q = \gamma^{DQN} = \gamma^{PPO} = 0.5$ with a

Algorithm 3 PPO for path planning

1: **Input:** field-recorded current velocity, continuous spatial positions, and PPO parameters;
2: **Output:** optimal PPO;
3: **for** each iteration **do**
4: Sample an initial state $s \leftarrow \{z, v_e\}$;
5: Run policy $\pi(.;\theta'^{PPO})$ over T and take $a \leftarrow z_e$;
6: Calculate advantage function estimates over T by (12) using R^{PPO} in (15);
7: Perform a gradient ascent on the surrogate function in (14);
8: Update $\theta'^{PPO} \leftarrow \theta^{PPO}$ every c_2 iterations;
9: **end for**

variance bias parameter of $\lambda = 0.9$ for PPO algorithm. In this paper, real ocean current velocity data from the Southeast Florida Gulf Stream measured by a 75 kHz ADCP at a latitude of 26.09°N and longitude of −79.80°E are used. It should be noted that we have trained our approach with large datasets (to consider the stochastic nature of ocean currents) for effective generalization.

B. Comparative Results

It is noteworthy to mention that the performance of the RL algorithm has already been justified with the existing methods, such as the MPC algorithm and A* algorithm, in the authors' previous works [14], [24], [29]. Three proposed reward functions are verified through testing in different RL algorithms, as shown in Fig. 2. To validate the performance of the proposed reward function, the results are presented in cumulative reward and cumulative harnessed energy during the training phase. The superior reward function for each algorithm should show a decent convergence in cumulative reward and tend towards the maximum energy. In the QL algorithm, the QL reward function and DQN reward function demonstrate acceptable performance, while the QL reward function is slightly better in terms of harnessed energy. All reward functions perform well for the DQN algorithm, while the convergence and harnessed energy for the DQN reward function beat other reward functions. For the PPO algorithm, the DQN and PPO reward functions surpass the QL reward function in convergence, where the latter can harness the maximum energy.

Also, the convergence results for QL, DQN, and PPO algorithms along with training episodes, are represented in Fig. 2, showing a different number of episodes to complete the training for each algorithm. For example, the PPO training needs more episodes than the DQN due to the larger action and state spaces in the continuous space, where the training episode takes 6.7 s for DQN and 0.107 s for PPO. Hence, the PPO is significantly faster than the DQN for the training per episode, but it needs almost three times more episodes to be fully trained. It should be noted that different values for the reward are gained by using different reward functions for QL (5), DQN (9)-(11), and PPO (15)-(17).

Fig. 3 illustrates the vertical positions found through the path planning, as well as the corresponding current velocity, net power, and cumulative energy harvested from the MCT. Two approaches are introduced as the baseline algorithms: (i) static MCT maintaining an equilibrium operating depth $z =$ 50 m, and (ii) A* algorithm as a nominee from classical path planning algorithms. The planned path by the Q-learning induces a globally optimal path due to the gridded environment precision, which is used as a baseline for the DQN algorithm as a representative of the deep RL algorithms for the gridded environment. Therefore, it is justified by the experimental results that the DQN is able to successfully find the optimal path (similar to the Q-learning except for one position at the time step of 68). Meanwhile, the PPO algorithm solves the path planning problem in a continuous environment facing a large set of feasible vertical positions, thereby increasing the complexity of the problem but upbringing a capability of larger space exploration to find a better optimal path than the discrete gridded environment. From the obtained vertical positions, the PPO follows almost a similar trend with the discrete approaches with increased precision in opting the positions.

The PPO algorithm outperforms the A* algorithm and the discrete algorithms in terms of current velocity and harvested power, resulting in cumulative energy of 34.585 MWh over 100 hour compared to the harvested energy in the case of A* (29.848 MWh), Q-learning (31.905 MWh), and DQN (31.822 MWh). It should be noted that applying spatiotemporal optimization and even classical path planning approaches increase the harvested energy than the static MCT with a total energy of 29.296 MWh.

Switching from a discrete environment to the continuous one leads to an intensive complexity but a higher accuracy in path planning, where caution should be taken to evaluate the required precision due to application and then make a selection between discrete and continuous approaches. For example, in our application, since the main objective is to maximize the cumulative harvested energy, where any little differences between various approaches (i.e., DQN and PPO) are intensified over time (especially in the real-time path planning for the MCT), it is important to select the approach resulting in the highest energy. Another important takeaway from this paper is that testing different reward functions is essential to reach the best performance from the RL algorithm. Although the results from this paper can be extended to other energy-harvesting autonomous vehicles, future works are required to narrow down some guidelines on how to choose an appropriate reward function according to the system (linear or non-linear), environment (continuous or discrete), and the type of RL algorithm.

C. Extension to Energy-Harvesting Autonomous Vehicles

The proposed framework can be generalized to other energy-harvesting autonomous vehicles (such as airborne wind energy) to employ the real-recorded data from the spatiotemporally varying environment for path planning. The following steps are required to extend to other applications:

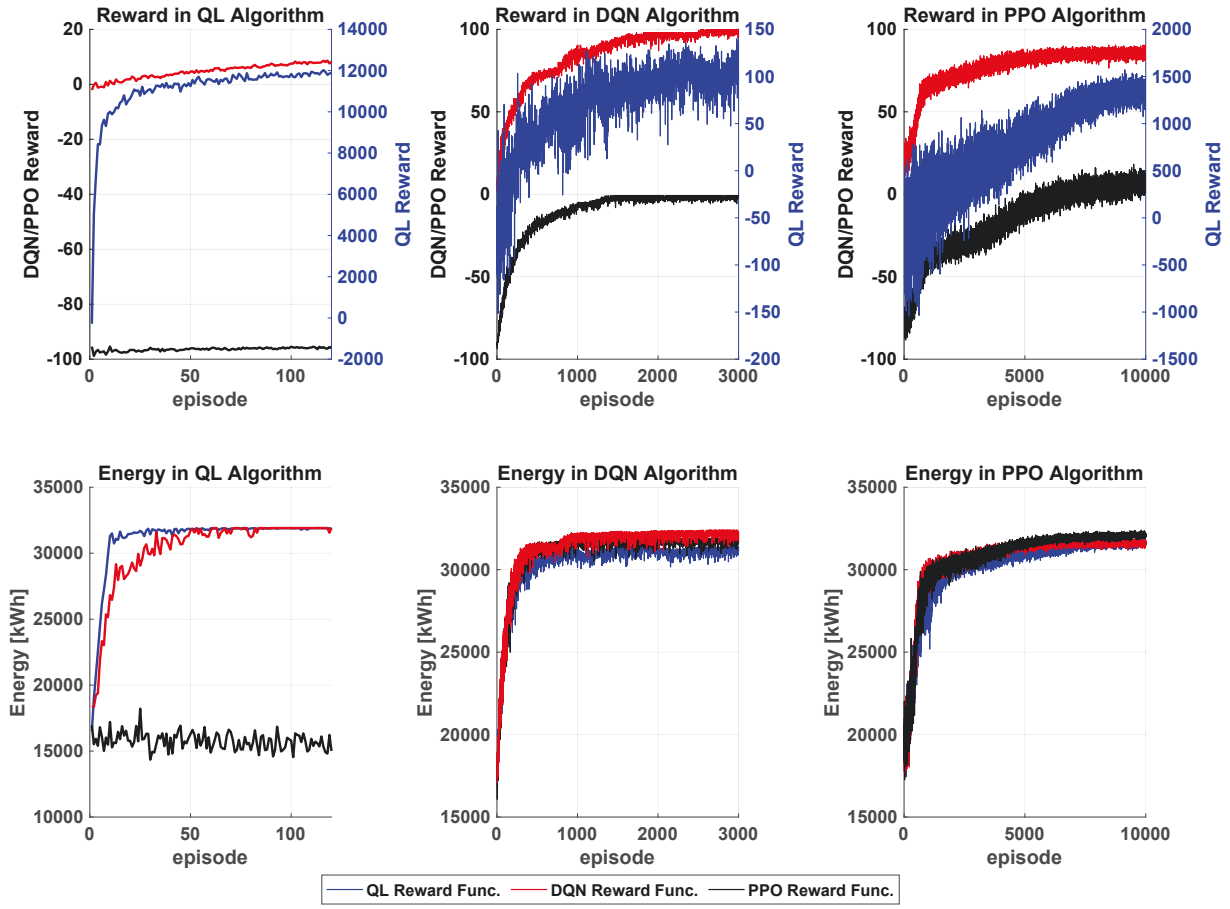

Fig. 2: Testing three proposed reward functions in QL algorithm, DQN algorithm, and PPO algorithm; (Top) Cumulative reward along with training episodes; (Bottom) Cumulative energy harnessed by MCT along with training episodes.

1) The ultimate objective is still power maximization; it is then intuitive to find an equation modeling the net power of the agent (similar to (2a)-(2d)).
2) The spatiotemporal environment can be modeled in either discrete gridded or continuous manners with respect to path accuracy. The key concern here is to access sufficient field-recorded data to train the RL algorithms.
3) The specific energy-harvesting vehicle should be adapted with the RL framework, where the states are defined according to the environment and agent's operating mechanism, and the action set introduces the potential paths.
4) The final step is to find the most suitable reward function to yield the optimal path with the highest harvested power. Three reward functions are introduced in this paper to cope with both discrete and continuous path planning, showing promising results and can be adjusted to other applications.

V. CONCLUSIONS

In this paper, an RL-based framework was presented to deal with the path planning of the AUV operating in the spatiotemporally varying underwater environment. For this framework, three RL algorithms were nominated to solve the path planning for both discrete and continuous representations from the environment and compare the experimental results. The investigated approach was tested on a case study of autonomous MCT while presenting the primary ideas on the extension to other energy-harvesting autonomous vehicles in similar uncertain environments.

REFERENCES

[1] J. S. Willners, L. Toohey, and Y. Petillot, "Sampling-based path planning for cooperative autonomous maritime vehicles to reduce uncertainty in range-only localization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3987–3994, 2019.

[2] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 418–429, 2004.

[3] J. Reed, J. Daniels, A. Siddiqui, M. Cobb, and C. Vermillion, "Optimal exploration and charging for an autonomous underwater vehicle with energy-harvesting kite," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4134–4139.

[4] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4889–4895.

[5] D. Saccani and L. Fagiano, "Autonomous uav navigation in an unknown environment via multi-trajectory model predictive control,"

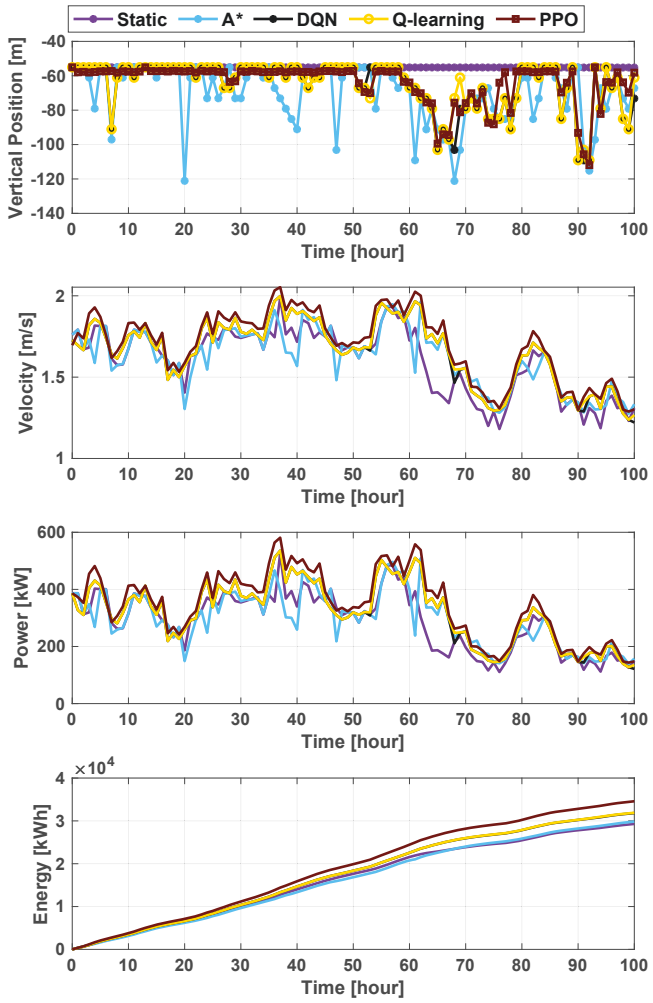

Fig. 3: Comparative results (vertical position z, velocity v_e, power P_{net}, and energy E_{net}) obtained by static MCT, A*, Q-learning, DQN, and PPO algorithms.

in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1577–1582.

[6] M. K. Cobb, K. Barton, H. Fathy, and C. Vermillion, "Iterative learning-based path optimization for repetitive path planning, with application to 3-d crosswind flight of airborne wind energy systems," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1447–1459, 2019.

[7] A. Bafandeh and C. Vermillion, "Real-time altitude optimization of airborne wind energy systems using lyapunov-based switched extremum seeking control," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 4990–4995.

[8] S. Bin-Karim, M. Muglia, and C. Vermillion, "Centralized position optimization of multiple agents in spatiotemporally-varying environment: a case study with relocatable energy-harvesting autonomous underwater vehicles in the gulf stream," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2019, pp. 264–269.

[9] A. Hasankhani, J. VanZwieten, Y. Tang, B. Dunlap, A. De Luera, C. Sultan, and N. Xiros, "Modeling and numerical simulation of a buoyancy controlled ocean current turbine," *International Marine Energy Journal*, vol. 4, no. 2, pp. 47–58, 2021.

[10] A. Hasankhani, Y. Tang, A. Snyder, J. VanZwieten, and W. Qiao, "Control co-design for buoyancy-controlled mhk turbine: A nested optimization of geometry and spatial-temporal path planning," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2022, pp. 1159–1165.

[11] K. P. Carroll, S. R. McClaran, E. L. Nelson, D. M. Barnett, D. K. Friesen, and G. N. William, "Auv path planning: an a* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones," in *Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology*. IEEE, 1992, pp. 79–84.

[12] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field d* algorithm," *Journal of Field Robotics*, vol. 23, no. 2, pp. 79–101, 2006.

[13] S. Bin-Karim, A. Bafandeh, A. Baheri, and C. Vermillion, "Spatiotemporal optimization through gaussian process-based model predictive control: A case study in airborne wind energy," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 798–805, 2017.

[14] A. Hasankhani, Y. Tang, J. VanZwieten, and C. Sultan, "Comparison of deep reinforcement learning and model predictive control for real-time depth optimization of a lifting surface controlled ocean current turbine," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2021, pp. 301–308.

[15] R. Gieselmann and F. T. Pokorny, "Planning-augmented hierarchical reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5097–5104, 2021.

[16] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2016.

[17] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, 2008.

[18] Z. Wang, S. Zhang, X. Feng, and Y. Sui, "Autonomous underwater vehicle path planning based on actor-multi-critic reinforcement learning," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 235, no. 10, pp. 1787–1796, 2021.

[19] Y. Sun, J. Cheng, G. Zhang, and H. Xu, "Mapless motion planning system for an autonomous underwater vehicle using policy gradient-based deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3, pp. 591–601, 2019.

[20] Z. He, L. Dong, C. Sun, and J. Wang, "Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[21] P. Bhopale, F. Kazi, and N. Singh, "Reinforcement learning based obstacle avoidance for autonomous underwater vehicle," *Journal of Marine Science and Application*, vol. 18, no. 2, pp. 228–238, 2019.

[22] B. Sangiovanni, G. P. Incremona, M. Piastra, and A. Ferrara, "Self-configuring robot path planning with obstacle avoidance via deep reinforcement learning," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 397–402, 2020.

[23] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved q-learning for path planning of a mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1141–1153, 2013.

[24] A. Hasankhani, Y. Tang, J. VanZwieten, and C. Sultan, "Spatiotemporal optimization for vertical path planning of an ocean current turbine," *IEEE Transactions on Control Systems Technology*, 2022.

[25] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, "Autonomous uav navigation: A ddpg-based deep reinforcement learning approach," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[26] A. Hasankhani, T. B. Ondes, Y. Tang, C. Sultan, and J. Van Zwieten, "Integrated path planning and tracking control of marine current turbine in uncertain ocean environments," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3106–3113.

[27] R. S. Sutton, A. G. Barto *et al.*, "Introduction to reinforcement learning," 1998.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[29] A. Hasankhani, Y. Tang, Y. Huang, and J. Van Zwieten, "Real-time vertical path planning using model predictive control for an autonomous marine current turbine," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2022, pp. 1166–1171.