Mapless Navigation of a Hybrid Aerial Underwater Vehicle with Deep Reinforcement Learning Through Environmental Generalization

1st Ricardo B. Grando
Technological University of Uruguay
Technological University of Uruguay
Rivera, Uruguay
ricardo.bedin@utec.edu.uy

2nd Junior C. de Jesus
Centro de Ciências Computacionais
Universidade Federal de Rio Grande
Rio Grande, Brazil
dranaju@gmail.com

3rd Victor A. Kich
Universidade Federal de Santa Maria
Universidade Federal de Santa Maria (UFSM)
Santa Maria, Brazil
victorkich@yahoo.com.br

4th Alisson H. Kolling
Universidade Federal de Santa Maria
Universidade Federal de Santa Maria (UFSM)
Santa Maria, Brazil
alikolling@gmail.com

5th Pedro M. Pinheiro
Centro de Ciências Computacionais
Universidade Federal de Rio Grande
Rio Grande, Brazil
pedrompinheiro@hotmail.com

6th Rodrigo S. Guerra
Centro de Ciências Computacionais
Universidade Federal de Rio Grande
Rio Grande, Brazil
rodrigo.guerra@furg.br

7th Paulo L. J. Drews-Jr
Centro de Ciências Computacionais
Universidade Federal de Rio Grande
Rio Grande, Brazil
paulodrews@furg.br

***Abstract*—Previous works showed that Deep-RL can be applied to perform mapless navigation, including the medium transition of Hybrid Unmanned Aerial Underwater Vehicles (HUAUVs). This paper presents new approaches based on the state-of-the-art actor-critic algorithms to address the navigation and medium transition problems for a HUAUV. We show that a double critic Deep-RL with Recurrent Neural Networks improves the navigation performance of HUAUVs using solely range data and relative localization. Our Deep-RL approaches achieved better navigation and transitioning capabilities with a solid generalization of learning through distinct simulated scenarios, outperforming previous approaches.**

Supplementary Material

Video of the experiments are available at: `https://youtu.be/rKqUMOKzgSI`. Released code at: `https://github.com/ricardoGrando/DoCRL`.

I. Introduction

Several studies about Hybrid Unmanned Aerial Underwater Vehicles (HUAUVs) have been published recently [1]–[8]. These types of vehicles enable an interesting range of new applications due to their capability to operate both in the air and underwater. These include inspection and mapping of partly submerged areas in industrial facilities, search and rescue and others. Most of the literature in the field is still focused on vehicle design, with few published works on the theme of autonomous navigation [9]. The ability to navigate in both environments and successfully transit from one to another imposes additional challenges that must be addressed.

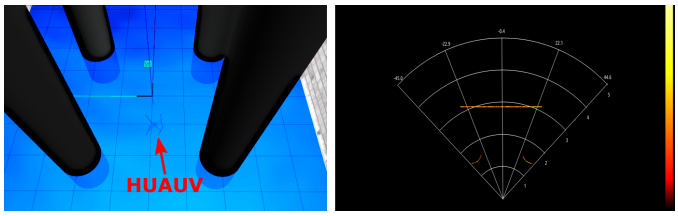

Fig. 1: Our HUAUV underwater in the first scenario (left) and its respective sonar readings (right).

Lately, approaches based on Deep-RL have been enhanced to address navigation-related tasks for a range of mobile vehicles, including ground mobile robots [10], aerial robots [11], [12] and underwater robots [13]. Based on actor-critic methods and multi-layer network structures, these approaches have achieved interesting results in mapless navigation, obstacle avoidance, even including media transitioning for HUAUVs [9], [14]. However, the challenges faced by this kind of vehicle make these existing approaches still too limited, with poor generalization through different scenarios.

In this work, we present two new double-critic Deep-RL approaches in the context of HUAUVs to perform navigation-related tasks in a continuous state-space environment: (1) a deterministic approach based on Twin Delayed Deep Determin-

978-1-6654-6280-8/22/$31.00 ©2022 IEEE

istic Policy Gradient (TD3) [15]; and (2) a stochastic approach based on Soft Actor-Critic (SAC) [16]. We show we are capable of training agents that are consistently better than state-of-the-art in generalizing through different simulated scenarios, with improved stability in mapless navigation, obstacle avoidance and medium transitions. Our evaluation tasks included both air-to-water and water-to-air transitions. We compared our methods with other single critic approaches and with an adapted version of a traditional Behavior-Based Algorithm (BBA) [17] used in aerial vehicles.

This work provides the following main contributions:

- We show that our agents present a robust capacity for generalization through different environments, achieving a good performance in a complex and completely unknown environment. The robot also performs the medium transition, being capable of arriving at the desired target and avoiding collisions.
- We show that a Long Short Term Memory (LSTM) architecture can achieve better overall performance and capacity for generalization than the state-of-the-art Multi-Layer Perceptron (MLP) architectures.

This work has the following structure: the related works are discussed in the following section (Sec. II). Following it, we present our methodology in Sec. III. The results are presented in Sec. IV and discussed in Sec. V.

II. Related Work

For more traditional types of vehicles, several works have been published demonstrating how efficiently Deep-RL can solve the mapless navigation problem [18]. For a ground robot, Tai *et al.* [19] demonstrated a mapless motion planner based on the DDPG algorithm employing a 10-dimensional range finder combined with the relative distance to the target as inputs and continuous steering signals as outputs. Recently, Deep-RL methods have also been successfully used by Ota *et al.* [10], de Jesus *et al.* [20], [21] and others, to accomplish mapless navigation-related tasks for terrestrial mobile robots. Singh and Thongam [22] demonstrated efficient near-optimal navigation for a ground robot in dynamic environments employing an MLP to perform speed control while choosing collision-free path segments.

For UAVs, Kelchtermans and Tuytelaars [23] demonstrated how memory could help Deep Neural Networks (DNN) for navigation in a simulated room crossing task. Tong *et al.* [11] showed better than state-of-the-art convergence and effectiveness in adopting a DRL-based method combined with a LSTM to navigate a UAV in highly dynamic environments, with numerous obstacles moving fast.

When it comes to problems involving specifically mapless navigation for UAVs, few works examine the effectiveness of Deep-RL. Grando *et al.* [24] explored a Deep-RL architecture, however, navigation was constrained to a 2D space. Rodriguez *et al.* [25] employed a DDPG-based strategy to solve the problem of landing UAVs on a moving platform. Similar to our work, they employed RotorS framework [26] combined with the Gazebo simulator. Sampedro *et al.* [27] proposed a DDPG-based strategy for search and rescue missions in indoor environments, utilizing real and simulated visual data. Kang *et al.* [28] also used visual information, although he focused on the subject of collision avoidance. In a go-to-target task, Barros *et al.* [29] applied a SAC-based method for the low-level control of a UAV. Double critic-based Deep-RL approaches similar to the one proposed here have also been shown to yield good results [12].

The HUAUV literature is still mostly concerned with vehicle design and modeling [1]–[7]. Two works have recently tackled the navigation problem with the medium transition of HUAUVs [30], [9]. Pinheiro *et al.* [30] focused on smoothing the medium transition problem in a simulated model on MATLAB. Grando *et al.* [9] developed Deep-RL actor-critic approaches and a MLP architecture. These two works were developed using generic distance sensing information for aerial and underwater navigation. In contrast, our work relies on more realistic sensing data, with the simulated LIDAR and sonar being both based on real-world devices.

The HUAUV presented in this paper is based on Drews-Jr *et al.* [1] model, which Neto *et al.* [2] has largely expanded. Our work differs from the previously discussed works by only using the vehicle's relative localization data and not its explicit localization data. We also present Deep-RL approaches based on double critic techniques instead of single critic, with RNN structures instead of MLP, traditionally used for mapless navigation of mobile robots. We compare our approaches with state-of-the-art Deep-RL approaches and with a behavior-based algorithm [17] adapted for hybrid vehicles to show that our new methodology improves the overall capability to generalize through distinct environments.

III. Methodology

In this section, we describe our simulation environment, our hybrid vehicle, and the proposed Deep-RL, detailing the network structure for both deterministic and stochastic agents. We also introduce the task that the vehicle must accomplish autonomously and the respective reward function.

A. Deterministic Deep RL

Developing on the DQN [31], Deep Deterministic Policy Gradient (DDPG) [32] employs an actor-network where the output is a vector of real values representing the chosen action, and a second neural network to learn the target function, providing stability and making it ideal for mobile robots [20]. While it provides good results, DDPG still has its problems, like overestimating the Q-values, which leads to policy breaking. TD3 [15] uses DDPG as its backbone, adding some improvements, such as clipped double-Q learning with two neural networks as targets for the Bellman error loss functions, delayed policy updates, and Gaussian noise on the target action, raising its performance.

Our deterministic approach is based on the TD3 technique. The pseudocode can be seen in Algorithm 1.

We train for max_steps steps in max_eps episodes. Our approach starts by exploring random actions for the initial

Algorithm 1 Deep Reinforcement Learning Deterministic

Initialize params of critic networks θ_1, θ_2 , and actor network ϕ
Initialize params of target networks $\phi' \leftarrow \phi$, $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$
Initialize replay buffer β
for $ep = 1$ to max_eps **do**
  reset environment state
  for $t = 0$ to max_steps **do**
    if $t < start\_steps$ **then**
      $a_t \leftarrow$ env.action_space.sample()
    else
      $a_t \leftarrow \mu_\phi(s_t) + \epsilon,\ \epsilon \sim \mathcal{N}(0, OU)$
    end if
    $s_{t+1},\ r_t,\ d_t,\ \_ \leftarrow$ env.step(a_t)
    store the new transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ into β
    if $t > start\_steps$ **then**
      Sample mini-batch B of N transitions $(s_t, a_t, r_t, s_{t+1}, d_t)$ from β
      $a' \leftarrow \mu_{\phi'}(s') + \epsilon,\ \epsilon \sim clip(\mathcal{N}(0, \tilde{\sigma}), -c,\ c)$
      Computes target:
      $Q_t \leftarrow r + \gamma * \min_{i=1,2} Q_{\theta_i}(s', a')$
      Update double critics with one step gradient descent:
      $\nabla_{\theta_i} \frac{1}{N} \sum_{i \in B} (Q_t - Q_{\theta_i(s_t, a_t)})^2$ for i=1,2
      if t % $policy\_freq(t)$ == 0 **then**
        Update policy with one step gradient descent:
        $\nabla_\phi \frac{1}{N} \sum_i [\nabla_{a_t} Q_{\theta_1}(s_t, a_t)|_{a_t = \mu(\phi)} \nabla_\phi \mu_\phi(s_t)]$
        Soft update for the target networks:
        $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
        $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$ for i=1,2
      end if
    end if
  end for
end for

$start\_steps$ steps. We use an LSTM as the actor-network ϕ and ϕ' as its target. The double critics are also LSTM networks, denoted by θ_1 and θ_2, with θ'_1 and θ'_2 as their targets. The learning of both networks happens simultaneously, addressing approximation error, reducing the bias, and finding the highest Q-values. The actor target chooses the action a' based on the state s', and we add Ornstein-Uhlenbeck noise to it. The double critic targets take the tuple (s', a') and return two Q-values as outputs, from which only the minimum of the two is considered. The loss is calculated with the Mean Squared Error of the approximate value from the target networks and the value from the critic networks. We use Adaptive Moment Estimation (Adam) to minimize the loss.

We update the policy network less frequently than the value network, taking into account a $policy\_freq$ factor that increases over time by the following rule:

$$policy\_freq(t) = \left\lfloor \left(0.5 - \frac{t}{max\_steps \times 3}\right)^{-1} \right\rfloor$$

B. Stochastic Deep RL

We also introduce a bias-stochastic actor-critic algorithm based on SAC [16], that combines off-policy updates with a stochastic actor-critic method to learn continuous action space policies. It uses neural networks as approximation functions to learn a policy and two Q-values functions similarly to TD3. However, SAC utilizes the current stochastic policy to act without noise, providing better stability and performance, maximizing the reward and the policy's entropy, encouraging the agent to explore new states and improving training speed. We use the soft Bellman equation with neural networks as a function approximation to maximize entropy. The pseudocode can be seen in Algorithm 2.

Algorithm 2 Deep Reinforcement Learning Stochastic

Initialize params of critic networks θ_1, θ_2 , and actor network ϕ
Initialize params of target networks $\phi' \leftarrow \phi$, $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$
Initialize replay buffer β
for $ep = 1$ to max_eps **do**
  reset environment state
  for $t = 0$ to max_steps **do**
    if $t < start\_steps$ **then**
      $a_t \leftarrow$ env.action_space.sample()
    else
      $a_t \leftarrow$ sample from $\pi_\phi(\cdot|s_t)$
    end if
    $s_{t+1},\ r_t,\ d_t,\ \_ \leftarrow$ env.step(a_t)
    store the new transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ into β
    if $t > start\_steps$ **then**
      Sample mini-batch B of N transitions $(s_t, a_t, r_t, s_{t+1}, d_t)$ from β
      $\tilde{a}_t \leftarrow$ sample from $\pi_\phi(\cdot|s_t)$
      $double = ([min_{i=1,2}(Q_{\theta'_i}(s_t, \tilde{a}_t)) - \alpha \log \tilde{a}_t)])$
      $Q_t = r(s_t, a_t) + \gamma(1 - d_t) * double$
      Update double critics with one step gradient descent:
      $\nabla_{\theta_i} \frac{1}{N} \sum_{s_t \in B} (Q_t - Q_{\theta_i}(s_t, a_t))^2$ for $i = 1, 2$
      if t % $policy\_freq(t)$ == 0 **then**
        Update policy with one step gradient descent:
        $\nabla_\phi \frac{1}{N} \sum_{s_t \in B} ([min_{i=1,2}(Q_{\theta_i}(s_t, \tilde{a}_t)) - \alpha \log \tilde{a}_t])$
        Soft update for the target networks:
        $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
        $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$ for i=1,2
      end if
    end if
  end for
end for

Like before, here we train for (max_steps) steps in (max_eps) episodes as well, exploring random actions for the first ($start\_steps$) steps. An LSTM structure was used for the policy network ϕ. After sampling a batch B from the memory β, we compute the targets for the Q-functions $Q_t(r_t, s_{t+1}, d_t)$, and update the Q-functions. Also, here we update the policy less frequently than the value network, using the same $policy\_freq$ factor we used in our deterministic approach.

C. Simulated Environments

Our experiments were conducted on the Gazebo simulator together with ROS, using the RotorS framework [26] to allow the simulation of aerial vehicles with different command levels, such as angular rates, attitude, location control and the simulation of wind with an Ornstein-Uhlenbeck noise. The underwater simulation is enabled by the UUV simulator [33], which allows the simulation of hydrostatic and hydrodynamic effects, as well as thrusters, sensors, and external perturbations. With this framework, we define the vehicle's underwater model with parameters such as the volume, additional mass, center of buoyancy, etc., as well as the characteristics of the underwater environment itself.

We developed two environments that simulate a walled water tank, with dimensions of $10\times10\times6$ meters and a one-meter water column. The first environment has four cylindrical columns representing subsea drilling risers. The second environment simulates complex structures, like those found in sea platforms, and contains several elements, such as walls, half walls and pipes.

D. HUAUV Description

Our vehicle was based on the model presented by Drews-Jr *et al.* [1], Neto *et al.* [2] and *et al.* [7], [34]. We described it using its actual mechanical settings, including inertia, motor coefficients, mass, rotor velocity, and others. A ROS package containing the vehicle's description plus the Deep-RL agents can be found in the Supplementary Material.

The vehicle sensing was optimized to mimic real-world LIDAR and Sonar. The described LIDAR is based on the UST 10LX model. It provides a 10 meters distance sensing with $270°$ of range and $0.25°$ of resolution, simulated using the plugin ray of Gazebo. Our simulated FLS sonar was based on the sonar simulation plugin developed by Cerqueira *et al.* [35]. We described a FLS sonar with 20 meters of range, with a bin count of 1000 and a beam count of 256. The width and height angles of the beam were $90°$ and $15°$, respectively. We obtained these values from the relative localization data using Rotors' geometric controller. In the real world, localization information can be obtained from a combination of standard localization sensing of hybrid vehicles like Global Positioning System (GPS) and Ultra Short Baseline (USBL).

E. Network Structure and Rewarding System

The structure of both our approaches has a total of 26 dimensions for the state, 20 samples for the distance sensors, the three previous actions and three values related to the target goal, which are the vehicle's relative position to the target and relative angles to the target in the x-y plane and the z-distance plane. When in the air, 20 samples come from the LIDAR. We get these samples equally spaced by $13.5°$ in the $270°$ LIDAR. When underwater, the distance information comes from the Sonar. We also get 20 beams equally spaced among the total of 256, and we take the highest bin in each beam. This conversion based on the range gives us the distance towards the obstacle or the tank's wall [36], [37]. The actions are scaled between 0 and $0.25\ m/s$ for the linear velocity, from $-0.25\ m/s$ to $0.25\ m/s$ for the altitude velocity and from -0.25 to $0.25\ rad$ for the Δ yaw.

1) Reward Function: We proposed a binary rewarding function that yields a positive reward in case of success or a negative reward in case of failure or in case the episode (ep) ends at the 500 steps limit:

$$r(s_t, a_t) = \begin{cases} r_{arrive} & \text{if } d_t < c_d \\ r_{collide} & \text{if } min_x < c_o \ || \ ep = 500 \end{cases} \quad (1)$$

The reward r_{arrive} was set to 100, while the negative reward $r_{collide}$ was set to -10. Both c_d and c_o distances were set to 0.5 meters.

IV. Experimental Results

In this section, the results obtained during our evaluation are shown. During the training phase, we created a randomly generated goal towards which the agent should navigate. The agents trained for a maximum of 500 steps or until they collided with an obstacle or with the tank's border. In case of reaching the goal before the limit of episodes, a new random goal was generated, allowing the total amount of reward to eventually exceed 100. A learning rate of 10^{-3} was used, with a minibatch of 256 samples and the Adam optimizer for all approaches, including the compared methods. We limited the number of episodes trained to 1500 episodes. The limits for the episode number (max_steps) were used based on the stagnation of the maximum average reward received.

For each scenario and model, an extensive amount of statistics were collected. The task addressed is goal-oriented navigation considering medium transition, where the robot must navigate from a starting point to an endpoint. This task was addressed in two ways in our tests: (1) starting in the air, performing the medium transition and navigating to a target underwater; and the other way around, (2) starting underwater, performing the medium transition and navigating to a target in the air. We collected the statistics for each of our proposed models (Det. and Sto.) and compared them with the performance of the state-of-the-art deterministic (Det.) and stochastic (Sto.) Deep-RL methods for HUAUVs, as well as a behavior-based algorithm [17] . These tasks were performed for 100 trials each and we recorded the total of successful trials, the average time for both underwater (t_water) and aerial (t_air) navigation and their standard deviations.

The models were all trained in the first environment and evaluated in both first (same as trained) and second (never seen) environments. We aim to outline one of the main contributions of this work, *i.e.* the robust capacity to generalize of our method across environments, in this case performing in a second, unknown and more complex environment. We set the initial position for the Air-Water (A-W) trials to (0.0, 0.0, 2.5) in the Gazebo Cartesian coordinates for the two scenarios. The target position used was (3.6, -2.4, -1.0). In both environments, the target was defined in a path with obstacles on the way. Table I shows the results obtained for each environment for 100 navigation trials.

We also performed a complementary comparison in the

TABLE I: Mean and standard deviation metrics over 100 navigation trials for all approaches in all scenarios.

Env	Test	t_{air} (s)	t_{water} (s)	Success
1	A-W Det.	76.28 ± 63.20	12.51 ± 20.71	94
1	A-W Sto.	21.79 ± 4.57	25.58 ± 5.70	100
1	A-W Sto. Grando *et al.* [9]	42.46 ± 62.94	13.13 ± 15.15	42
1	**A-W Det. Grando *et al.* [9]**	**13.84 ± 2.11**	**5.44 ± 1.73**	**100**
1	A-W BBA	32.42 ± 1.79	21.27 ± 0.18	100
1	W-A Det.	24.66 ± 10.06	5.0 ± 0.71	83
1	W-A Sto.	79.73 ± 27.91	5.41 ± 0.34	100
2	**A-W Det.**	61.94 ± 45.29	**8.44 ± 9.09**	**73**
2	**A-W Sto.**	**14.89 ± 1.120**	18.48 ± 6.24	**94**
2	A-W Sto. Grando *et al.* [9]	-	-	0
2	A-W Det. Grando *et al.* [9]	-	-	0
2	A-W BBA	39.69 ± 21.92	11.32 ± 7.46	28
2	**W-A Det.**	**8.54 ± 4.44**	**4.27 ± 0.47**	**8**
2	**W-A Sto.**	**15.43 ± 13.39**	**6.60 ± 1.75**	**10**
2	W-A Sto. Grando *et al.* [9]	-	-	0
2	W-A Det. Grando *et al.* [9]	-	-	0
2	W-A BBA	34.3 ± 22.93	6.13 ± 17.22	8

TABLE II: Mean and standard deviation metrics over 100 navigation trials tested in the second simulated environment, for both deterministic and stochastic models trained only in the first environment (Env1), in both first and second environments (Both), and only in the second environment (Env2).

Model	t_{air} (s)	t_{water} (s)	Success
A-W Det. (Env1)	61.94 ± 45.29	**8.44 ± 9.09**	**73**
A-W Sto. (Env1)	**14.89 ± 1.120**	18.48 ± 6.24	**94**
A-W Det. (Both)	**14.14 ± 3.77**	**8.69 ± 3.17**	**99**
A-W Sto. (Both)	16.82 ± 2.12	14.92 ± 3.60	**100**
A-W Det. (Env2)	23.17 ± 31.12	32.53 ± 60.28	21
A-W Sto. (Env2)	19.98 ± 15.99	49.61 ± 27.86	87
W-A Det. (Env1)	8.54 ± 4.44	4.27 ± 0.47	8
W-A Sto. (Env1)	15.43 ± 13.39	6.60 ± 1.75	10
W-A Det. (Both)	**25.09 ± 38.86**	**4.62 ± 0.51**	**34**
W-A Sto. (Both)	33.41 ± 11.82	11.40 ± 2.38	**83**
W-A Det. (Env2)	-	-	0
W-A Sto. (Env2)	3.73 ± 2.97	30.47 ± 9.47	1

second scenario. We used the models trained in the second environment to collect statistics. For a better analysis, we also performed a comparison between models in this second environment. First, we collected the data for Deterministic and Stochastic models trained only in the first environment for 1500 episodes (Env1), as shown before. Then, we trained these models for 500 more episodes in the second environment (Both). Lastly, we compared them with Deterministic and Stochastic trained only in the second environment for 1500 episodes. Table II shows the obtained results.

V. Conclusions

The evaluation shows an overall increase in performance in navigation through both environments. It is possible to see that our approaches achieve a consistent performance of 100 successful air-to-water navigation trials with also a consistent navigation time (14.55 ± 0.87 and 11.19 ± 2.86). In this same scenario, the stochastic performed a little worse in air-to-water navigation but outperformed the deterministic approach in water-to-air navigation. In the second scenario, we can see more clearly that a double-critic-based approach with an RNN structure also has a better ability to learn and generalize the environment, including the obstacles and the medium transition. While the state-of-the-art approaches with a MLP structure were not capable of performing the task, our approaches presented once again a consistent performance, especially in air-to-water navigation. Our approaches showed an excellent ability to learn the tasks and the environmental difficulties, not only the scenario itself. That was further addressed in our additional evaluation with agents trained in the first environment only, both first and second environments and the second environment only. Overall, we can conclude that double critic approaches with recurrent neural networks present a consistent ability to learn through scenarios and environments and to generalize between them. Also, our approaches outperformed the BBA algorithm in the rate of successful trials and average time in almost all situations.

It is important to mention that these approaches are extensively evaluated in a realistic simulation, including control issues and disturbances such as wind. Thus, the results indicate that our approach may achieve real-world application if the correct data from the sensing and the relative localization are correctly ensured. Finally, it is also possible to analyze that these new RNN-based approaches provided a more consistent average course of action throughout the environments.The evaluation shows an overall increase in performance in navigation through both environments. It is possible to see that our approaches achieve a consistent performance of 100 successful air-to-water navigation trials with also a consistent navigation time (14.55 ± 0.87 and 11.19 ± 2.86). In this same scenario, the stochastic performed a little worse in air-to-water navigation but outperformed the deterministic approach in water-to-air navigation. In the second scenario, we can see more clearly that a double-critic-based approach with an RNN structure also has a better ability to learn and generalize the environment, including the obstacles and the medium transition. While the state-of-the-art approaches with a MLP structure were not capable of performing the task, our approaches presented once again a consistent performance, especially in air-to-water navigation. Our approaches showed an excellent ability to learn the tasks and the environmental difficulties, not only the scenario itself. That was further addressed in our additional evaluation with agents trained in the first environment only, both first and second environments and the second environment only. Overall, we can conclude that double critic approaches with recurrent neural networks present a consistent ability to learn through scenarios and environments and to generalize between them. Also, our approaches outperformed the BBA algorithm in the rate of successful trials and average time in almost all situations.

It is important to mention that these approaches are extensively evaluated in a realistic simulation, including control issues and disturbances such as wind. Thus, the results indicate that our approach may achieve real-world application if the correct data from the sensing and the relative localization are correctly ensured. Finally, it is also possible to analyze that these new RNN-based approaches provided a more consistent average course of action throughout the environments.

By using physically realistic simulation in several water-tank-based scenarios, we showed that our approaches achieved an overall better capability to perform autonomous navigation, obstacle avoidance and medium transition than other approaches. Disturbances such as wind were successfully assimilated and good generalization through different scenarios was achieved. With our simple and realistic sensing approach that took into account only the range information, we presented overall better performance than the state-of-the-art and classical behavior-like algorithm. Future studies with our real HUAUV are on the way.

Acknowledgment

The authors would like to thank the VersusAI and NAUTEC team. This work was partly supported by the CAPES, CNPq and ANP-PRH 22.

REFERENCES

[1] P. L. Drews-Jr, A. A. Neto, and M. F. Campos, "Hybrid unmanned aerial underwater vehicle: Modeling and simulation," in *IEEE/RSJ IROS*, 2014, pp. 4637–4642.

[2] A. A. Neto, L. A. Mozelli, P. L. Drews-Jr, and M. F. Campos, "Attitude control for an hybrid unmanned aerial underwater vehicle: A robust switched strategy with global stability," in *IEEE ICRA*, 2015, pp. 395–400.

[3] R. T. da Rosa, P. J. Evald, P. L. Drews-Jr, A. A. Neto, A. C. Horn, R. Z. Azzolin, and S. S. Botelho, "A comparative study on sigma-point kalman filters for trajectory estimation of hybrid aerial-aquatic vehicles," in *IEEE/RSJ IROS*, 2018, pp. 7460–7465.

[4] M. M. Maia, D. A. Mercado, and F. J. Diez, "Design and implementation of multirotor aerial-underwater vehicles with experimental results," in *IEEE/RSJ IROS*, 2017, pp. 961–966.

[5] D. Lu, C. Xiong, Z. Zeng, and L. Lian, "A multimodal aerial underwater vehicle with extended endurance and capabilities," in *IEEE ICRA*, 2019, pp. 4674–4680.

[6] D. Mercado, M. Maia, and F. J. Diez, "Aerial-underwater systems, a new paradigm in unmanned vehicles," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 229–238, 2019.

[7] A. C. Horn, P. M. Pinheiro, R. B. Grando, C. B. da Silva, A. A. Neto, and P. L. Drews-Jr, "A novel concept for hybrid unmanned aerial underwater vehicles focused on aquatic performance," in *IEEE LARS/SBR*, 2020, pp. 1–6.

[8] V. M. Aoki, P. M. Pinheiro, P. L. J. Drews-Jr, M. A. B. Cunha, and L. G. Tuchtenhagen, "Analysis of a hybrid unmanned aerial underwater vehicle considering the environment transition," in *IEEE LARS/SBR*, 2021, pp. 90–95.

[9] R. B. Grando, J. C. de Jesus, V. A. Kich, A. H. Kolling, N. P. Bortoluzzi, P. M. Pinheiro, A. Alves Neto, and P. L. J. Drews-Jr, "Deep reinforcement learning for mapless navigation of a hybrid aerial underwater vehicle with medium transition," in *IEEE ICRA*, 2021, pp. 1088–1094.

[10] K. Ota, Y. Sasaki, D. K. Jha, Y. Yoshiyasu, and A. Kanezaki, "Efficient exploration in constrained environments with goal-oriented reference path," in *IEEE/RSJ IROS*, 2020, pp. 6061–6068.

[11] G. Tong, N. Jiang, L. Biyue, Z. Xi, W. Ya, and D. Wenbo, "UAV navigation in high dynamic environments: A deep reinforcement learning approach," *Chinese Journal of Aeronautics*, vol. 34, no. 2, pp. 479–489, 2021.

[12] R. B. Grando, J. C. de Jesus, V. A. Kich, A. H. Kolling, and P. L. J. Drews-Jr, "Double critic deep reinforcement learning for mapless 3d navigation of unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, pp. 1–14, 2022.

[13] I. Carlucho, M. De Paula, S. Wang, B. V. Menna, Y. R. Petillot, and G. G. Acosta, "Auv position tracking control using end-to-end deep reinforcement learning," in *MTS/IEEE OCEANS*, 2018.

[14] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, R. d. S. Guerra, and P. L. J. Drews Jr, "Depth-cuprl: Depth-imaged contrastive unsupervised prioritized representations in reinforcement learning for mapless navigation of unmanned aerial vehicles," *arXiv preprint arXiv:2206.15211*, 2022.

[15] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *ICML*, 2018, pp. 1587–1596.

[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, vol. 80, 2018, pp. 1861–1870.

[17] R. Marino, F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "A minimalistic quadrotor navigation strategy for indoor multi-floor scenarios," in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 1561–1570.

[18] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ IROS*, 2017.

[19] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *IEEE/RSJ IROS*, 2017, pp. 31–36.

[20] J. C. de Jesus, J. A. Bottega, M. A. Cuadros, and D. F. Gamarra, "Deep deterministic policy gradient for navigation of mobile robots in simulated environments," in *19th ICAR*, 2019, pp. 362–367.

[21] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, M. A. d. S. L. Cuadros, and D. F. T. Gamarra, "Soft actor-critic for navigation of mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, pp. 1–11, 2021.

[22] N. H. Singh and K. Thongam, "Mobile robot navigation using mlp-bp approaches in dynamic environments." *Arabian Journal for Science & Engineering*, vol. 43, p. 8013–8028, 2018.

[23] K. Kelchtermans and T. Tuytelaars, "How hard is it to cross the room?–training (recurrent) neural networks to steer a UAV," *arXiv preprint arXiv:1702.07600*, 2017.

[24] R. B. Grando, P. M. Pinheiro, N. P. Bortoluzzi, C. B. da Silva, O. F. Zauk, M. O. Piñeiro, V. M. Aoki, A. L. Kelbouscas, Y. B. Lima, P. L. Drews-Jr, and A. A. Neto, "Visual-based autonomous unmanned aerial vehicle for inspection in indoor environments," in *IEEE LARS/SBR*, 2020, pp. 1–6.

[25] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, I. G. Moreno, and P. Campoy, "A deep reinforcement learning technique for vision-based autonomous multirotor landing on a moving platform," in *IEEE/RSJ IROS*, 2018, pp. 1010–1017.

[26] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*, 2016, pp. 595–625.

[27] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente, and P. Campoy, "A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques," *Journal of Intelligent & Robotic Systems*, pp. 601–627, 2019.

[28] K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine, "Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight," in *IEEE ICRA*, 2019, pp. 6008–6014.

[29] G. M. Barros and E. L. Colombini, "Using Soft Actor-Critic for Low-Level UAV Control," *arXiv e-prints*, vol. abs/2010.02293, 2020.

[30] P. M. Pinheiro, A. A. Neto, R. B. Grando, C. B. da Silva, V. M. Aoki, D. Cardoso, A. C. Horn, and P. L. J. Drews-Jr, "Trajectory planning for hybrid unmanned aerial underwater vehicles with smooth media transition," *arXiv preprint arXiv:2112.13819*, 2021.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop*, 2013.

[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR*, 2016.

[33] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *MTS/IEEE OCEANS*, 2016, pp. 1–8.

[34] A. C. Horn, P. M. Pinheiro, C. B. Silva, A. A. Neto, and P. L. Drews-Jr, "A study on configuration of propellers for multirotor-like hybrid aerial-aquatic vehicles," in *19th ICAR*, 2019, pp. 173–178.

[35] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira, "A novel gpu-based sonar simulator for real-time applications," *Computers & Graphics*, vol. 68, pp. 66–76, 2017.

[36] M. M. Santos, P. Drews-Jr, P. Núñez, and S. Botelho, "Object recognition and semantic mapping for underwater vehicles using sonar data," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 2, pp. 279–289, 2018.

[37] M. M. Santos, G. B. Zaffari, P. O. C. S. Ribeiro, P. L. J. Drews-Jr, and S. S. C. Botelho, "Underwater place recognition using forward-looking sonar images: A topological approach," *Journal of Field Robotics*, vol. 36, no. 2, pp. 355–369, 2019.