

DoCRL: Double Critic Deep Reinforcement Learning for Mapless Navigation of a Hybrid Aerial Underwater Vehicle with Medium Transition

1st Ricardo B. Grando

Robotics Lab

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 Rodrigo S. Guerra

Centro de Ciências Computacionais

Universidade Federal de Rio Grande
Rio Grande, Brazil

rodrigo.guerra@furg.br

6th Paulo L. J. Drews-Jr

Centro de Ciências Computacionais

Universidade Federal de Rio Grande
Rio Grande, Brazil

paulodrews@furg.br

Abstract—Deep Reinforcement Learning (Deep-RL) techniques for motion control have been continuously used to deal with decision-making problems for a wide variety of robots. Previous works showed that Deep-RL can be applied to perform mapless navigation, including the medium transition of Hybrid Unmanned Aerial Underwater Vehicles (HUAUVs). These are robots that can operate in both air and water media, with future potential for rescue tasks in robotics. This paper presents new approaches based on the state-of-the-art Double Critic Actor-Critic algorithms to address the navigation and medium transition problems for a HUAUV. We show that double-critic Deep-RL with Recurrent Neural Networks using range data and relative localization solely improves the navigation performance of HUAUVs. Our DoCRL approaches achieved better navigation and transitioning capability, outperforming previous approaches.

SUPPLEMENTARY MATERIAL

Video of the experiments available at: <https://youtu.be/PqTDzsKjA9c>. Released code at: <https://github.com/ricardoGrando/DoCRL>.

I. INTRODUCTION

Several studies about Hybrid Unmanned Aerial Underwater Vehicles (HUAUVs) have been conducted recently [1]–[8]. These vehicles provide an interesting range of possible applications due to the capability to act in two different environments, including inspection and mapping of partly submerged areas in industrial facilities, search and rescue and other military-related applications. However, the state-of-the-art is yet focused on the vehicle design and structure, where even fewer studies around autonomous navigation have been conducted [9]. The ability to perform tasks in both environments and successfully transit between them imposes additional challenges that must be addressed to make this mobile vehicle autonomously feasible.

Approaches based on Deep Reinforcement Learning (Deep-RL) techniques 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

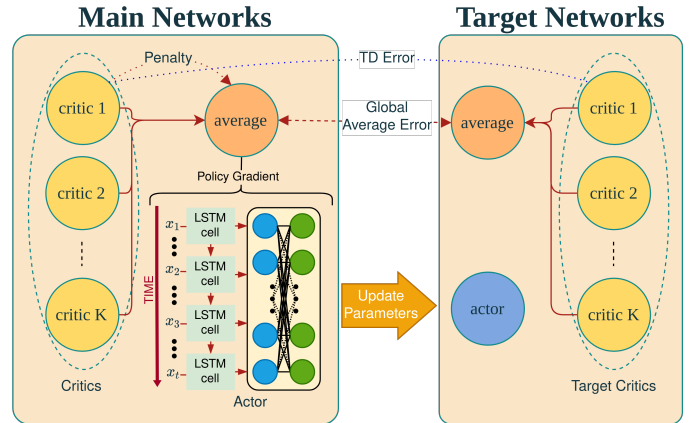


Fig. 1: DoCRL architecture.

robots [13]. These approaches based on single critic actor-critic techniques with multi-layer network structures have achieved interesting results in performing mapless navigation, obstacle avoidance and media transitioning even for HUAUVs [9]. However, the challenges faced by this kind of vehicle make these approaches limited, not being capable of escalating to more complex scenarios for a rescue navigation task, for example.

In this work, we explore the use of Deep-RL in the context of HUAUVs to perform navigation-related tasks that can simulate through environmental rescue tasks in robotics. We present two enhanced approaches based on state-of-the-art Deep-RL for the continuous state: (1) a deterministic based on Twin Delayed Deep Deterministic Policy Gradient (TD3) [14]; and (2) a stochastic based on Soft Actor-Critic (SAC) [15]. We show that we are capable of training agents with a consistently better capability than state-of-the-art, performing with more stability the mapless navigation, obstacle avoidance and medium transition. We perform a two-fold way evaluation

with air-to-water and water-to-air navigation. We compare our DoCRL approaches with single critic-based approaches used to perform mapless navigation and with an adapted version of a traditional Behavior-Based Algorithm (BBA) [16] used in aerial vehicles. Our proposed double critic formulation can be seen in Fig. 1.

This work contains the following main contributions:

- We propose two approaches based on state-of-the-art actor-critic double critic Deep-RL algorithms that can successfully perform goal-oriented mapless navigation for HUAUVs, using only range data readings and the vehicles' relative localization data.
- We show that a Long Short Term Memory (LSTM) architecture achieves better overall performance than the state-of-the-art Multi-Layer Perceptron (MLP) architecture.
- We show that our robot presents a robust capacity to navigate in scenarios that can simulate through environmental (air-water) rescue tasks in robotics. The robot also performs the medium transition, capable of arriving at the desired target and avoiding collisions.

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. Finally, we discuss our contributions and present future works in Sec. VI.

II. RELATED WORK

The HUAUV literature is still mostly concerned with mechanical design and modelling [1]–[7]. Autonomous navigation-related problems are still barely addressed for this kind of vehicle [9]. The HUAUV used in this paper [9] was created based on Drews-Jr *et al.* [1] model, which Neto *et al.* [2] has largely expanded.

Several Deep-RL works in robotics have previously been carried out for the mapless navigation problem, demonstrating how efficiently we may solve the problem utilizing learning techniques [17]. For a mapless motion planner of a ground robot, Tai *et al.* [18] employed 10-dimensional range findings and the relative distance of the vehicle to a target as inputs and continuous steering signals as outputs. According to the results, a mapless motion planner based on the DDPG algorithm may be effectively taught and utilized to navigate to a target. Recently, deep-RL methods have also been successfully used in robotics by Ota *et al.* [10], de Jesus *et al.* [19], [20], and others to accomplish mapless navigation-related tasks for terrestrial mobile robots.

Kelchtermans and Tuytelaars [21] used a LSTM to perform autonomous navigation in a UAV. The room crossing task performed by Kelchtermans and Tuytelaars approach demonstrated through simulation how memory can help Deep Neural Networks (DNN) in navigation control. Tong *et al.* [11] proposed a DRL-based method using a LSTM to navigate a UAV in high dynamic environments, with numerous obstacles moving fast. Their approach achieved superiority in terms of convergence and effectiveness compared with the state-of-the-art DRL methods. Singh and Thongam [22] employed a

Multi-Layer Perceptron (MLP) to perform terrestrial mobile robot navigation in dynamic environments. Their method is used to choose a collision-free segment and controls the robot's speed for each motion. They demonstrated that the method is efficient and gives a near-optimal path reaching the target position of the mobile robot.

When it comes to problems involving mapless navigation for Unmanned Aerial Vehicles, the effectiveness of Deep-RL is somewhat limited (UAVs) [23]. Rodriguez *et al.* [24] employed a DDPG-based strategy to solve the problem of landing on a moving platform. It employed Deep-RL in conjunction with the RotorS framework [25] to simulate UAVs in the Gazebo simulator. Sampedro *et al.* [26] proposed a DDPG-based strategy for the Search and Rescue mission in interior situations, utilizing visual data from a real and simulated UAV. Kang *et al.* [27] also used visual information, although he focused on the subject of collision avoidance. In a go-to-target task, Barros *et al.* [28] applied a SAC-based method to low-level control of a UAV. Grando *et al.* [29] utilized approaches based on the DDPG and SAC algorithms on Gazebo for 2D UAV navigation. Recently, double critic-based Deep-RL approaches have been developed for UAVs, presenting better results [12].

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 approaches following single critic structure and a MLP architecture. These two works were developed using generic distance sensing information for aerial and underwater navigation. In contrast, more sophisticated sensing with a real-world simulated LIDAR and sonar is adopted in the present work.

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 DoCRL approaches with state-of-the-art Deep-RL approaches and with a behavior-based algorithm [16] adapted for hybrid vehicles to show that our methodology improves the overall capability to perform autonomous navigation.

III. METHODOLOGY

In this section, we discuss our Deep-RL approaches. We detail the network structure for both deterministic and stochastic agents. We also present the proposed reward function for the task that the vehicle must accomplish autonomously.

A. Double Critic Deep Reinforcement Learning Deterministic - DoCRL-D

Developing on the DQN [31], Deep Deterministic Policy Gradient (DDPG) [32] employs an actor-network where output is a real value that represents a chosen action, and a second neural network to learn the target function providing stability and making it ideal for mobile robots [19]. While providing

good results, DDPG still has its problems, like overestimating the Q-values leading to policy breaking. TD3 [14] uses DDPG as its backbone and improves it by 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 of DoCRL-D can be seen in Algorithm 1.

Algorithm 1 Double Critic Deep Reinforcement Learning Deterministic - DoCRL-D

```

1: Initialize params of critic networks  $\theta_1, \theta_2$ , and actor network  $\pi(\phi)$ 
2: Initialize params of target networks  $\phi' \leftarrow \phi, \theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$ 
3: Initialize replay buffer  $\beta$ 
4: for  $ep = 1$  to  $max\_eps$  do
5:   reset environment state
6:   for  $t = 0$  to  $max\_steps$  do
7:     if  $t < start\_steps$  then
8:        $a_t \leftarrow env.action\_space.sample()$ 
9:     else
10:       $a_t \leftarrow \pi_\phi(s) + \epsilon, \epsilon \sim \mathcal{N}(0, OU)$ 
11:    end if
12:     $s_{t+1}, r_t, d_t, \_ \leftarrow env.step(a_t)$ 
13:    store the new transition  $(s_t, a_t, r_t, s_{t+1}, d_t)$  into  $\beta$ 
14:    if  $t > start\_steps$  then
15:      Sample mini-batch of  $N$  transitions  $(s_t, a_t, r_t, s_{t+1}, d_t)$  from  $\beta$ 
16:       $a' \leftarrow \pi_{\phi'}(s') + \epsilon, \epsilon \sim clip(\mathcal{N}(0, \bar{\sigma}), -c, c)$ 
17:      Computes target:
18:       $Q_t \leftarrow r + \gamma * \min_{i=1,2} Q_{\theta_i}(s', a')$ 
19:      Update double critics with one step gradient descent:
20:       $\nabla_{\theta_i} \frac{1}{N} \sum_i (Q_t - Q_{\theta_i}(s_t, a_t))^2$  for  $i=1,2$ 
21:      if  $t \% policy\_freq == 0$  then
22:        Update policy with one step gradient descent:
23:         $\nabla_{\phi} \frac{1}{N} \sum_i [\nabla_{a_t} Q_{\theta_1}(s_t, a_t)|_{a_t=\pi(\phi)} \nabla_{\phi} \pi_{\phi}(s_t)]$ 
24:        Soft update for the target networks:
25:         $\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$ 
26:         $\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$  for  $i=1,2$ 
27:      end if
28:    end if
29:  end for
30: end for

```

We train for a number max steps (max_steps) in a number episodes (max_steps). Our approach starts by exploring random actions while a minimum number of steps ($start_steps$) has not been achieved yet.

We use an LSTM as actor-network and denote them by ϕ and its copy ϕ' as actor target. The double critic as well, by θ_1, θ_2 for parameterization of two value networks and its copies θ'_1, θ'_2 as critic targets. The learning of both networks happens simultaneously, addressing approximation error, reducing the bias, and finding the highest Q-value. 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 output. The minimum of the two target Q-values is considered as the approximated value return. 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 delay by updating the policy network less frequently than the value network. It is updated taking into account a $policy_freq$ factor that increases over time by the following rule:

$$policy_freq = (int) \frac{1}{0.5 - \frac{t}{max_steps \times 3}}$$

B. Double Critic Deep Reinforcement Learning Stochastic - DoCRL-S

A stochastic approach is also developed in this work. Our approach is based on the SAC algorithm [15]. This algorithm consists of a bias-stochastic actor-critic 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. It maximizes 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 the entropy. The pseudocode of DoCRL-S can be seen in Algorithm 2.

Algorithm 2 Double Critic Deep Reinforcement Learning Stochastic - DoCRL-S

```

1: Initialize critic networks  $\theta_1, \theta_2$ , and actor network  $\pi(\phi)$ 
2: Initialize target networks  $\phi' \leftarrow \phi, \theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$ 
3: Initialize replay buffer  $\beta$ 
4: for  $ep = 1$  to  $max\_eps$  do
5:   reset environment state
6:   for  $t = 0$  to  $max\_steps$  do
7:     if  $t < start\_steps$  then
8:        $a_t = env.action\_space.sample()$ 
9:     else
10:       $a_t \leftarrow \pi_\phi(\cdot|s)$ 
11:    end if
12:     $s_{t+1}, r_t, d_t, \_ \leftarrow env.step(a_t)$ 
13:    store the new transition  $S(s_t, a_t, r_t, s_{t+1}, d_t)$  into  $\beta$ 
14:    if  $t > start\_steps$  then
15:      Sample m-batch of  $N$  transitions  $(s_t, a_t, r_t, s_{t+1}, d_t)$  from  $\beta$ 
16:       $double = ([\min_{i=1,2} (Q_{\theta'_i}(s_t, a_t)) - \alpha \log \pi(a_t|s_t)])$ 
17:       $Q_t = r(s_t, a_t) + \gamma(1 - d_t) * double$ 
18:      Update double critics with one step gradient descent:
19:       $\nabla_{\theta_i} argmin_{\theta_i} \frac{1}{|N|} \sum (Q_t - Q_{\theta_i}(s_t, a_t))^2$  for  $i=1,2$ 
20:      if  $t \% policy\_freq == 0$  then
21:        Update policy with one step gradient descent:
22:         $\nabla_{\phi} \frac{1}{|N|} \sum_{s_t \in \beta} ([\min_{i=1,2} (Q_{\theta'_i}(s_t, a_{t,\phi})) - \alpha \log \pi(a_{t,\phi}|s_t)])$ 
23:        for  $i=1,2$ 
24:        Soft update for the target networks:
25:         $\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$  for  $i=1,2$ 
26:      end if
27:    end if
28:  end for
29: end for

```

We train for a number max steps (max_steps) in a number of episodes (max_steps) as well. It explores random actions while a minimum number of steps ($start_steps$) has not been achieved yet. A LSTM structure was used for the policy network ϕ . After sampling a batch from the memory, we compute the targets for the Q-functions $Q_t(r_t, s_{t+1}, d_t)$, and update the Q-functions. We perform the delay by updating the policy less frequently than the value network, as performed with our DoCRL-D approach. It was updated taking into account the same $policy_freq$ factor as the DoCRL-D.

C. Simulated Environments

Our experiments are conducted with the vehicle and on the second and more complex environment provided by Grando et al. [9]. Based on the Gazebo simulator together with ROS, it makes the use of the framework RotorS [25], which provides the means to simulate aerial vehicles with different command levels, such as angular rates, attitude, location control and the simulation of wind with an Ornstein-Uhlenbeck noise method. 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, the vehicle’s underwater model was defined with parameters such as the volume, additional mass, center of buoyancy, etc., as well as the characteristics of the underwater environment itself.

The environment simulate a walled water tank, with dimensions of $10 \times 10 \times 6$ meters and a one-meter water column. It has four cylindrical columns representing drilling risers.

1) *HUAUV Description*: The vehicle used is based on the model presented by Drews-Jr et al. [1], Neto et al. [2] and et al. [34]. It was described using its actual mechanics settings, including the inertia, motor coefficients, mass, rotor velocity, and others. Its sensing was optimized for real-world LIDAR and Sonar. The described LIDAR is based on the UST 10LX model. It provides a 10 meters distance sensing with 270° range and 0.25° of resolution. It was simulated using the plugin ray of Gazebo. The simulated FLS sonar is based on the sonar simulation plugin developed by Cerqueira et al. [35]. It has 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.

2) *Network Structure and Rewarding System*: The structure of 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 plan. When in the air, the 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 got 20 beams equally spaced among the total of 256, and we took for 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.

3) *Reward Function*: We proposed a binary rewarding function that takes into account 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:

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

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

meters.

IV. EXPERIMENTAL RESULTS

During the training phase, we created a randomly generated goal that the agent should navigate towards. The agents train for a maximum of 500 steps or until they collide with an obstacle or with the tank border. In case of reaching the goal before the limit of episodes, a new random goal was generated. In this case, the total amount of reward could exceed the maximum value of 100. It was used a learning rate of 10^{-3} , a minibatch of 256 samples and the Adam optimizer for all approaches, including for the compared methods. We limited the number of episodes of 1500 episodes. These respective limits for the episode number (max_steps) are used based on the stagnation of the maximum average reward received.

A. Results

In this section, the results obtained during our evaluation are shown. An extensive amount of statistics are 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 a two-fold way in our tests: starting in the air, performing the medium transition and navigating to a target underwater; and the other way around, starting underwater, performing the medium transition and navigating to a target in the air. We collected the statistics for each of our proposed models (DoCRL-D and DoCRL-S) 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 [16]. This two-fold task was performed for 100 trials and the total of successful trials are recorded. Also, the average time for both underwater (t_{water}) and aerial (t_{air}) along navigation with their standard deviations are recorded.

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 three scenarios. The target position was set to (2.0, 3.0, -1.0) at the bottom of the water tank. For the Water-Air (W-A) evaluation, the coordinates were swapped. The same was done for the second and third scenarios, where the coordinates used were (0.0, 0.0, 2.5) and (3.6, -2.4, -1.0). The target was defined in a path with obstacles on the way.

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

Test	t_{air} (s)	t_{water} (s)	Success
A-W DoCRL-D	14.55 ± 0.87	11.19 ± 2.86	100
A-W DoCRL-S	72.02 ± 24.33	9.97 ± 6.05	70
A-W Sto. Grando et al. [9]	18.10 ± 7.91	1.08 ± 2.38	1
A-W Det. Grando et al. [9]	34.09 ± 23.85	13.06 ± 12.98	14
A-W BBA	34.70 ± 30.22	4.87 ± 6.16	11
W-A DoCRL-D	16.67 ± 10.44	4.65 ± 0.69	42
W-A DoCRL-S	8.77 ± 7.23	5.98 ± 1.90	71
W-A Sto. Grando et al. [9]	15.33 ± 23.75	14.9 ± 18.88	12
W-A Det. Grando et al. [9]	27.58 ± 14.77	12.65 ± 7.89	13
W-A BBA	38.01 ± 29.83	3.79 ± 0.27	32

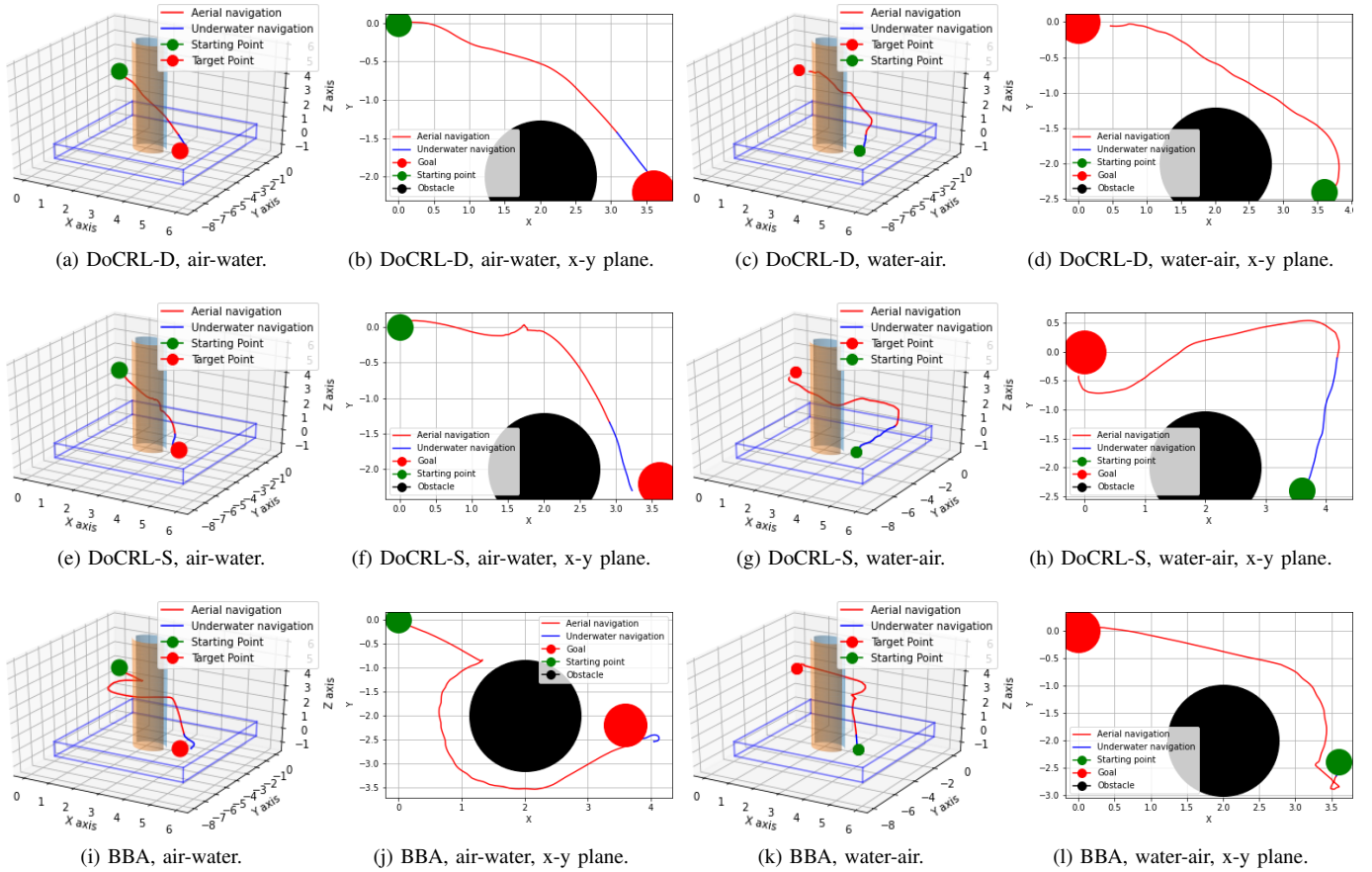


Fig. 2: Behavior sample of each approach tested in the environment trained. (figs. 2a to 2l).

V. DISCUSSION

The evaluation shows an overall increase in performance in navigation through the the scenario. We can see that the DoCRL-D approach achieves a consistent performance of 100 successfully 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 DoCRL-S approach performed a little worse in air-to-water navigation but outperformed the deterministic approach in water-to-air navigation.

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.

VI. CONCLUSIONS

In this paper, we presented two novels Deep-RL approaches based on RNNs and double critic structures to perform the navigation of a HUAUV. 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 a classical behavior-like algorithm. Our work aims to contribute to future studies focusing on rescue missions in robotics, particularly with vehicles like the HUAUV, and extending to environmental rescue tasks in both air and water domains. Furthermore, we are actively conducting research to use our HUAUV model to explore its potential in practical real world applications.

ACKNOWLEDGMENT

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

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] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *ICML*, 2018, pp. 1587–1596.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [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] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] K. Kang, S. Belkhal, 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.
- [28] G. M. Barros and E. L. Colomini, "Using Soft Actor-Critic for Low-Level UAV Control," *arXiv e-prints*, vol. abs/2010.02293, 2020.
- [29] R. B. Grando, J. C. de Jesus, and P. L. Drews-Jr, "Deep reinforcement learning for mapless navigation of unmanned aerial vehicles," in *IEEE LARS/SBR*, 2020, pp. 1–6.
- [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.