MKM511E- Special Topics in Mechatronics Engineering

Project Presentation

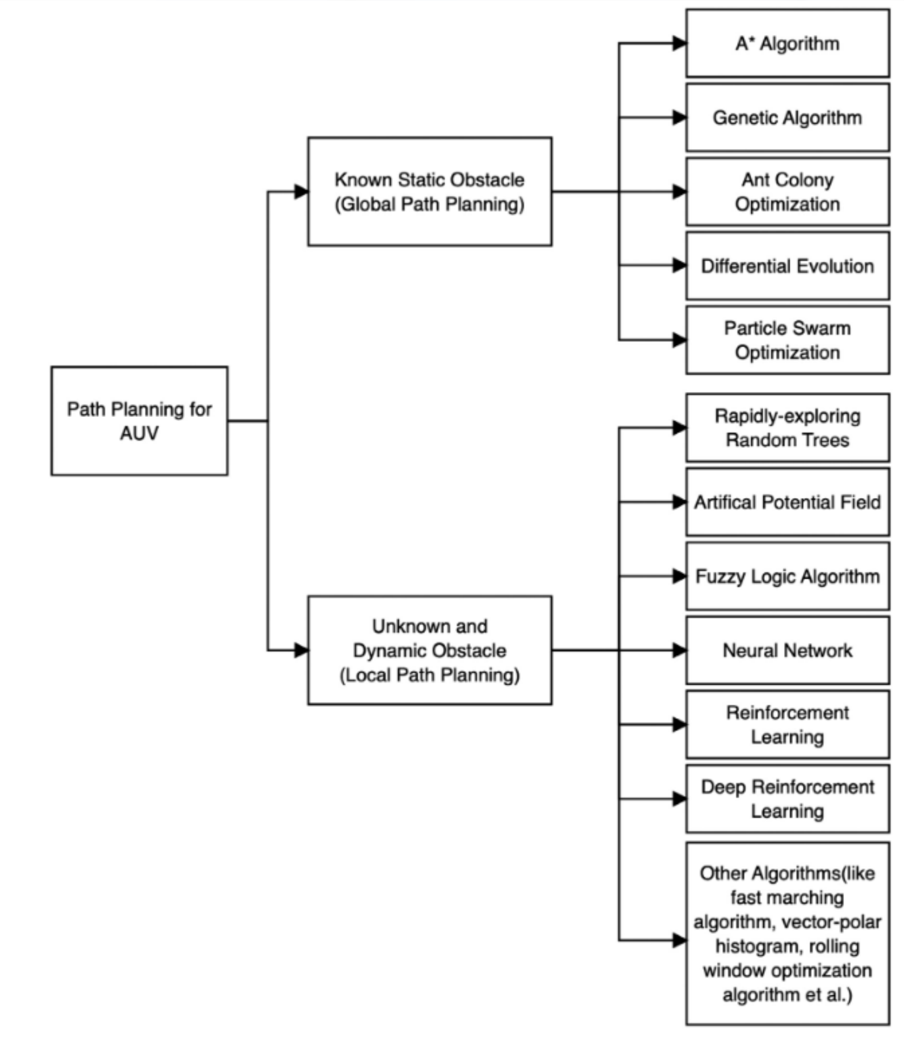
07.06.2023

Ferhad Kaleli

Background

Project Title: Developing 2D Path Planning for AUV Based on Deep RL

Background: Autonomous underwater vehicles are significant in marine exploration tasks. To realize autonomy, path planning and obstacle avoidance is the core technology. Path planning algorithms should work in the constraints and characteristics of AUV and the complex and changeable marine environment.
Path planning algorithms are divided into two groups mainly; global path planning with known static obstacles and local path planning with unknown and dynamic obstacles.



Main path planning algorithms for AUV [1]

Ferhad Kaleli

Problem Definition

Problem Definition: Improve the safe navigation of autonomous underwater vehicle with sonar sensors in a two-dimensional dynamic environment.

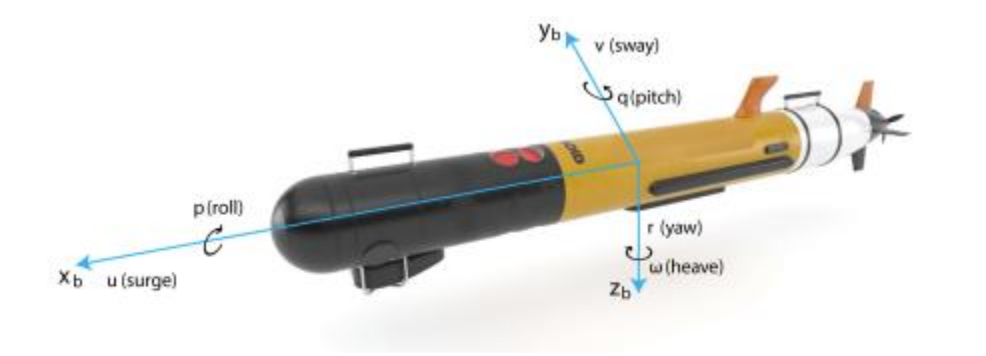


Figure 1. The earth-fixed and body-fixed frame of an AUV.

Table 1. The notation of 6-DOF states.

DOF	Forces and moments	Positions and Euler angles	Linear and angular velocities
Surge	X	x	u
Sway	Y	y	ν
Heave	Z	z	ω
Roll	K	ϕ	ρ
Pitch	M	θ	q
Yaw	N	ψ	r

Modelling of AUV

Modelling of AUV: The horizontal motion of the AUV with 3-DOF with motion components, surge, sway and yaw are considered.

$$\dot{\eta} = R(\psi)v$$

$$\dot{\psi} = r$$

$$\text{where } \eta := [x, y, \psi]^{\tau} \quad v := [u, v, r]^{\tau}$$

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

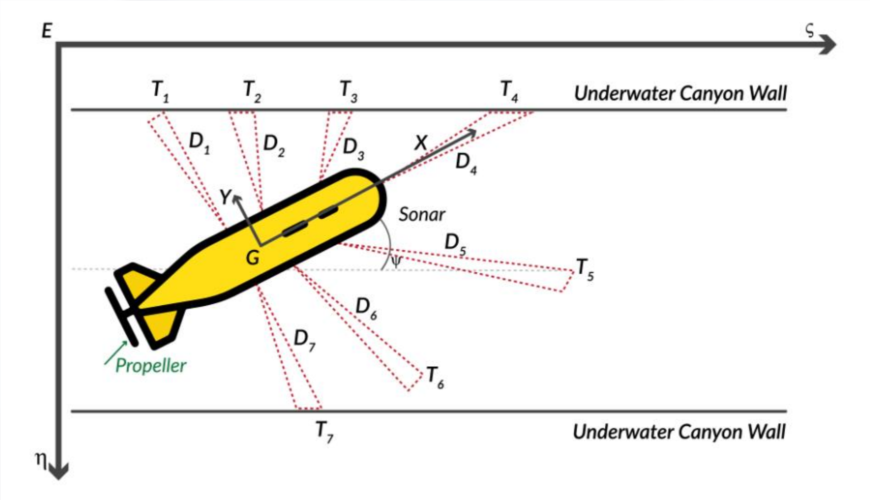


Figure 2. The geometric relationship of AUV facing continuous obstacles

Obstacle Avoidance Strategy:

- Large-scale continuous obstacles
- Dynamic obstacles

Environment Definition: 1000*300 m

- Task
  - Find optimal path to reach target (green box)
- States
  - 7 sonar distance sensors (150 m)

State	Sonar Detection Results	Range of Values
s_t^1	sonars 1 $D_1(t)$	$[0, 150]$
s_t^2	sonars 2 $D_2(t)$	$[0, 150]$
s_t^3	sonars 3 $D_3(t)$	$[0, 150]$
s_t^4	sonars 4 $D_4(t)$	$[0, 150]$
s_t^5	sonars 5 $D_5(t)$	$[0, 150]$
s_t^6	sonars 6 $D_6(t)$	$[0, 150]$
s_t^7	sonars 7 $D_7(t)$	$[0, 150]$

- Actions
  - Yaw angular velocity (-1, +1 rad/s)
  - Horizontal velocity (-1, +1.5 m/s)

$$a_t\{a_t^1, a_t^2\} = \{\omega(t), v(t)\}$$

- Algorithm
  - Proximal Policy Optimization (PPO)
- Reward Function Design

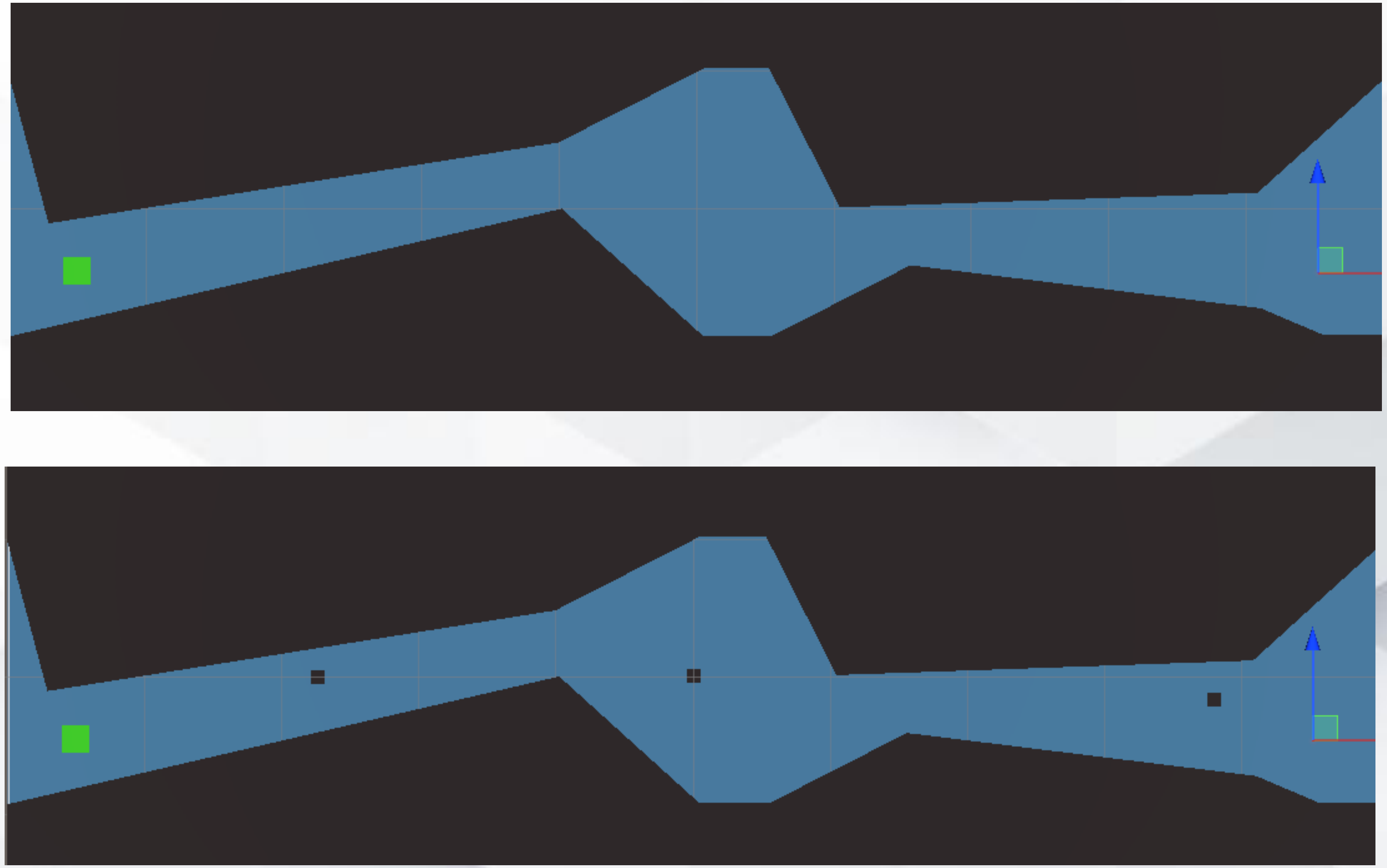


Figure 3. Environment Design in Unity

Ferhad Kaleli

ITÜ 250th YEAR 1773–2023

Algorithm

Proximal Policy Optimization (PPO)

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, \ldots$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

Compute rewards-to-go $\hat{R}_t$.

Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.

Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \;\; g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.

Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.

end for

Figure 5. Pseudocode of PPO [3]


```yaml
behaviors:
  Auv:
    trainer_type: ppo
    hyperparameters:
      batch_size: 512
      buffer_size: 10240
      learning_rate: 0.0003
      beta: 0.005
      epsilon: 0.2
      lambd: 0.99
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: true
      hidden_units: 256
      num_layers: 3
      vis_encode_type: simple
    reward_signals:
      extrinsic:
        gamma: 0.995
        strength: 1.0
    keep_checkpoints: 5
    max_steps: 1e6
    time_horizon: 128
    summary_freq: 12000
    threaded: True
```

Reward Function Design

Reward Function:

- Target module reward

$$r_1(s_t, a_t, s_{t+1}) = -0.001 \times \sqrt{\left(x_t - x_{goal}\right)^2 + \left(y_t - y_{goal}\right)^2}$$

- Safety module reward (R2 is based on max-step)

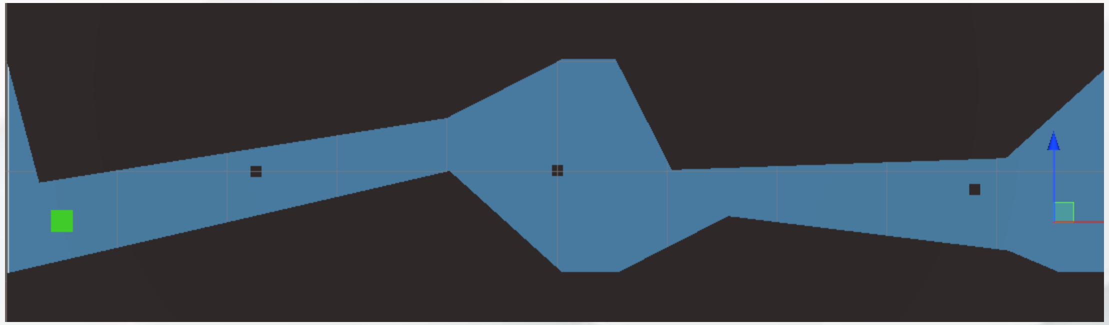
$$r_2^1(s_t, a_t, s_{t+1}) = \begin{cases} -R_2 & \text{if } \min(D_i(t)) \leq 1.0r_s (i = 1, 2, 3, \cdots 7) \\ -0.01 \times (\min(D_i(t)) - r_s)^2 & \text{if } \min(D_i(t)) \leq 2.0r_s (i = 1, 2, 3, \cdots 7) \\ 0 & \text{if } \min(D_i(t)) 2.0r_s (i = 1, 2, 3, \cdots 7) \end{cases}$$

- Stability reward

$$r_3(s_t, a_t, s_{t+1}) = -0.01 \times (|\omega_{t+1} - \omega_t|) + |v_{t+1} - v_T|)$$

- Reward function

$$r_3(s_t, a_t, s_{t+1}) = \tau_1 r_1(s_t, a_t, s_{t+1}) + \tau_2 r_2(s_t, a_t, s_{t+1}) + \tau_3 r_3(s_t, a_t, s_{t+1})$$



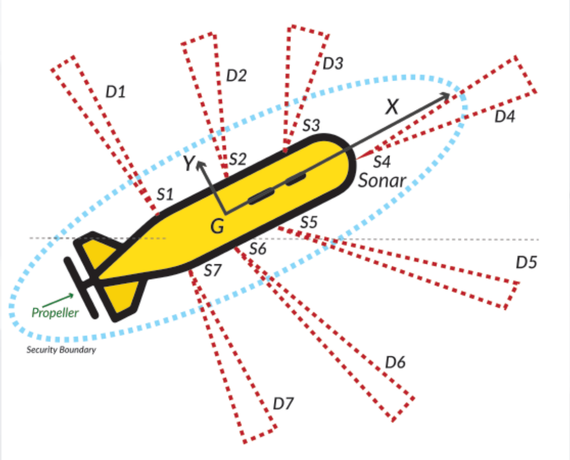


Figure 4. Safety Region of AUV.

Ferhad Kaleli

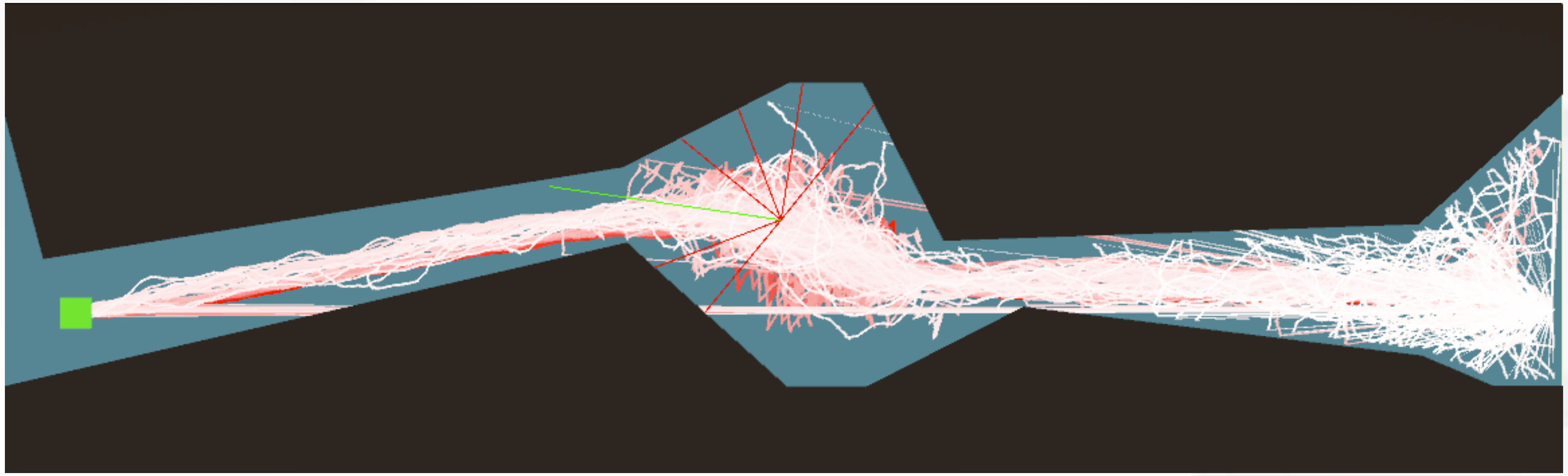


Figure 6. Simulation with PPO

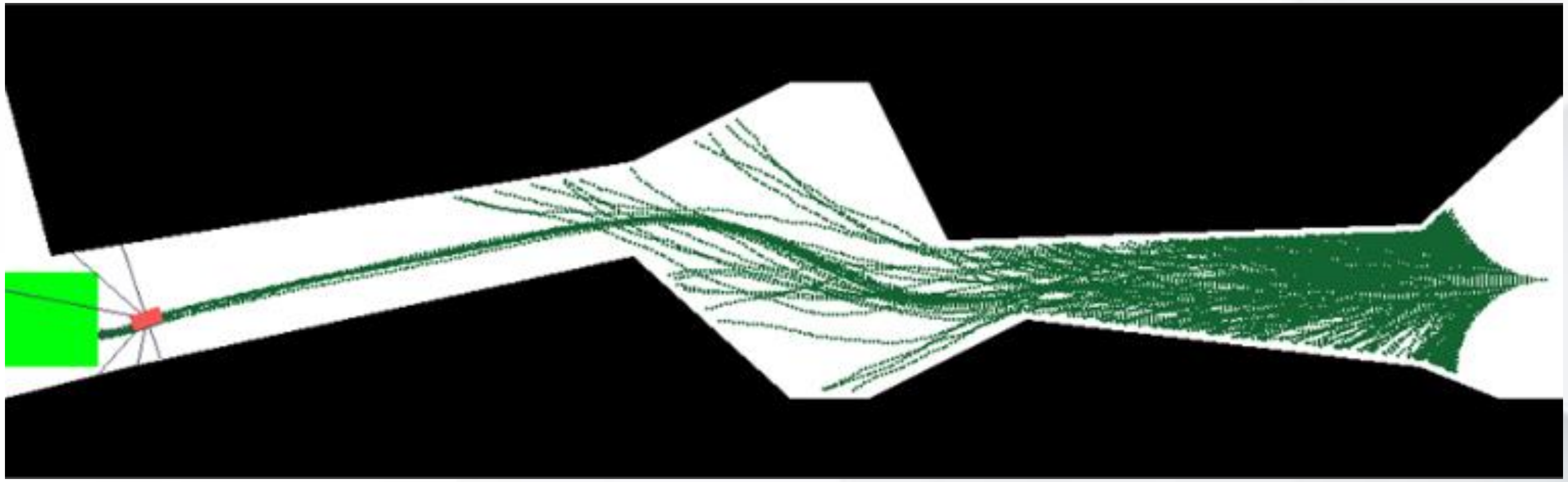


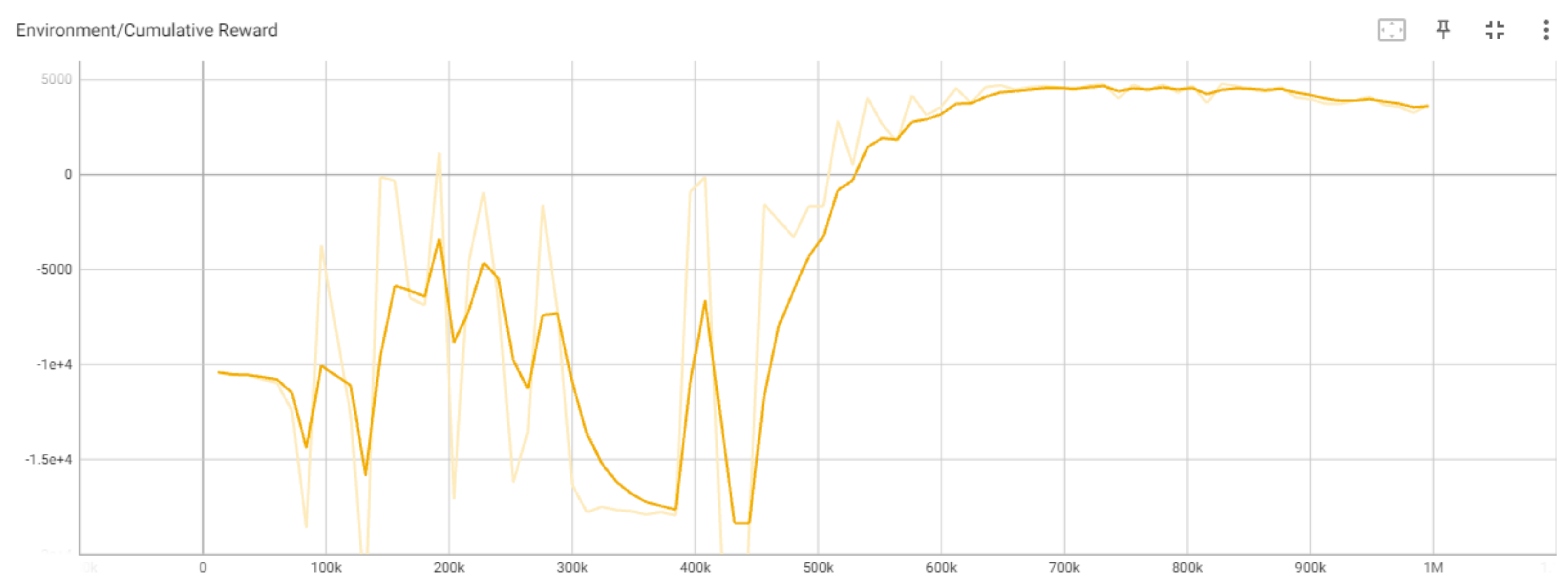
Figure 7. Simulation with Sum-Tree DDPG [2]

Simulation Environment:
Unity Machine Learning Agents Toolkit,
2 simulation scenarios

- Static Obstacle Environment
- Static Obstacle Environment with the addition of random static obstacles

Figure 6. Example Dynamic Environment [2]

Analysis



Algorithm	# Successful Hits to Target	Min-Episode of Convergence	Optimal Path (m)
DDPG	1 (1000)	791	1263.5
Sum-DDPG	218 (1000)	796	1128.5
PPO	**77 (166)**	**78**	**1096.9**

Ferhad Kaleli

Simulations

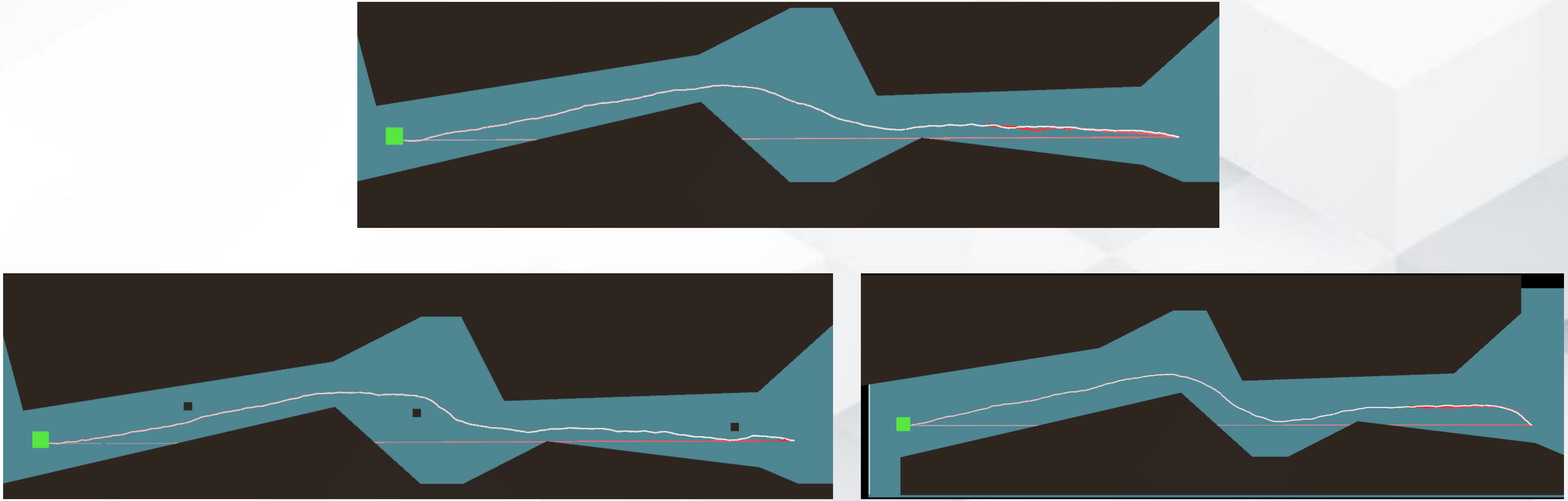


Figure 6. Example Dynamic Environment [2]

References

[1] Chunxi Cheng, Qixin Sha, Bo He, Guangliang Li, Path planning and obstacle avoidance for AUV: A review, Ocean Engineering, Volume 235, 2021, 109355, ISSN 0029-8018, https://doi.org/10.1016/j.oceaneng.2021.109355.
[2] Sun, Y.; Luo, X.; Ran, X.; Zhang, G. A 2D Optimal Path Planning Algorithm for Autonomous Underwater Vehicle Driving in Unknown Underwater Canyons. J. Mar. Sci. Eng. 2021, 9, 252. https:// doi.org/10.3390/jmse9030252
[3] OpenAI.com. Available online: https://spinningup.openai.com/en/latest/al.htgorithms/ppo.htmlpseudocode (26.04.2023).

Thank you

07.06.2023

Ferhad Kaleli