Introduction to Reinforcement Learning for Torpedo Guidance with Obstacle Avoidance

Hyunsoo Lee and Joongheon Kim
Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
E-mail: hyunsoo@korea.ac.kr, joongheon@korea.ac.kr

***Abstract*—This paper explores the application of reinforcement learning (RL) to torpedo guidance with a focus on obstacle avoidance and target acquisition in dynamic environments. By employing a dual-actor network approach and a regularization technique for the critic network, the proposed method demonstrates robust performance in scenarios with random obstacle placements and target directions. Our results show that the system effectively navigates toward the target while avoiding obstacles, leveraging reinforcement learning to optimize torpedo trajectories. The study addresses the challenges of overestimation and underestimation biases common in reinforcement learning methods, such as deep deterministic policy gradient (DDPG), by utilizing the double actors and regularized critics (DARC) algorithm. This approach has potential implications for enhancing the autonomy and efficiency of military applications in complex and uncertain settings.**

I. Introduction

Reinforcement learning (RL) is a branch of artificial intelligence (AI) in which an agent interacts with an environment through a sequence of decisions. The primary goal of reinforcement learning is to guide the agent to take actions that maximize the cumulative expected reward over time, thereby achieving optimal outcomes. Recent advancements in the field of RL have been crucial in solving various complex decision-making problems [1]–[3].

Particularly in the military domain, the application of AI has shown significant effectiveness in protecting friendly forces and efficiently neutralizing adversaries at a lower cost [4]. Warfare systems, such as weapons, navigation, and sensors, can employ AI to make tasks more efficient and less dependent on human input. In many cases, AI systems can rapidly and effectively analyze situations and make optimal decisions in critical moments.

In this paper, we conducted a study on torpedo guidance based on reinforcement learning, focusing on obstacle avoidance and target acquisition. By using two actor networks and applying a regularization technique to the critic network, we achieved stable performance even when the positions of obstacles and the target's direction were random [5].

This paper is structured as follows: In Sec. II, we provide an overview of reinforcement learning and discuss the recent technological trends in applying reinforcement learning to obstacle avoidance and path planning. In Sec. III, we describe our system model and the problem in detail, followed by a discussion of the implementation results. Finally, we conclude in Sec. IV.

II. Preliminaries

A. Reinforcement Learning

RL focuses on training an agent to learn an optimal policy in a given environment. The agent interacts with the environment by observing its state, selecting actions accordingly, and receiving rewards as a result. RL aims to learn a policy that maximizes the long-term cumulative reward through these interactions.

Reinforcement learning is generally divided into model-based and model-free approaches. In the model-based approach, the agent learns or is provided with a dynamic model of the environment, which it uses to predict future states and rewards, thereby designing an optimal policy. In contrast, the model-free approach does not involve directly learning the environment model; instead, the agent learns the optimal policy or value function directly from experience. The key components of reinforcement learning are about the agent's and environment's interaction. Agent means the learner or decision-maker observing the environment to achieve a goal. *State* (s) I the information that the agent can observe from the environment. The state represents the situation at a specific point in time within the environment. *Actions* (a) are the choices available to the agent in a particular state. Each action allows the agent to transition to the next state by receiving a *reward* (r). The reward is the feedback that the agent receives from the environment due to taking a specific action. The reward indicates how desirable the action was, and the primary goal of RL is to maximize this reward. *Policy* (π) is the strategy that determines the action the agent will take given a particular state. The policy can be either probabilistic or deterministic. To obtain the expected value of the cumulative reward that can be obtained by starting from a specific state and following the optimal policy, the *value function* ($V(s)$) is used to assess how advantageous it is for the agent to be in a particular state. Similar to the value function, the expected value of the cumulative reward when taking a specific action in a specific state can be obtained from *Q-Function*. The Q-function evaluates state-action pairs and plays a crucial role in selecting the optimal action.

Reinforcement learning algorithms are broadly categorized into value-based, policy-based, and actor-critic approaches. Q-learning, a value-based method, learns the Q-function to derive the optimal policy, while policy gradient methods, such as the policy gradient algorithm, directly learn the policy itself. The

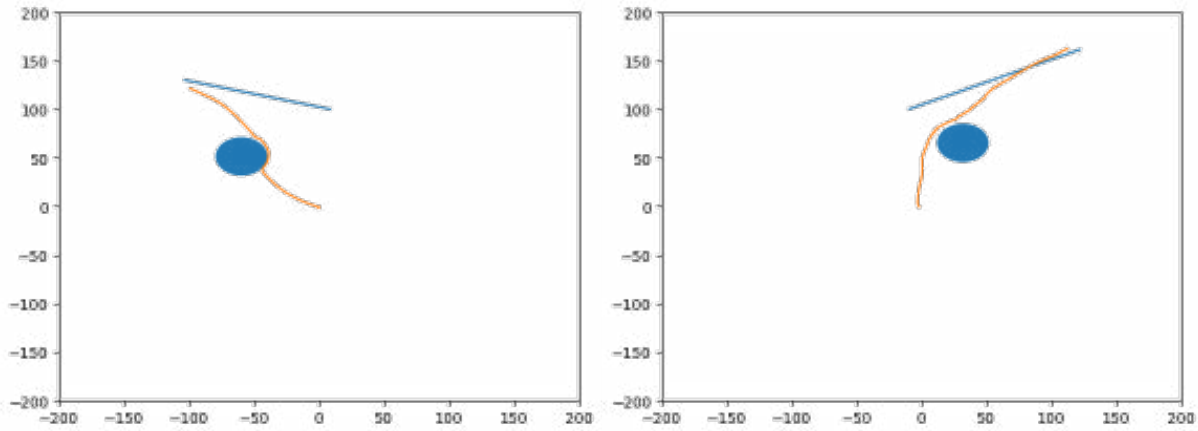

Fig. 1: Torpedo avoids obstacles and heads towards the target

actor-critic method combines policy-based and value-based approaches, simultaneously learning the policy and the value function. Among them, the actor-critic method has recently become RL's most widely used algorithm, combining policy-based and value-based approaches. The *actor* represents the policy that determines the actions to take in a given state. The actor learns a policy function $\pi_\theta(s)$, where θ represents the parameters of the policy function, which takes the state s as input and outputs an action a. The actor's goal is to update the policy to maximize the cumulative reward through interactions with the environment. Using policy gradient methods, the actor updates the policy in the direction that increases the expected cumulative reward under the current policy. The *critic* evaluates the actions taken by the actor. It learns a Q-function or value function $V(s)$ for the given state s and the action a chosen by the actor. Through this, the critic assesses how good the current policy is and provides feedback to the actor to update the policy. During the learning process, the critic approximates the Q-function or value function using the Bellman equation. The value calculated by the critic is crucial information for the actor to improve the policy.

B. Related Work

Advantage actor-critic (A2C) uses an advantage function instead of temporal-difference (TD) error to evaluate how much better the current action is compared to the average, updating the policy based on this evaluation [6]. A2C is more stable than conventional policy gradient methods because it uses the advantage function, which reduces the variance in the gradient estimates. However, It still has computational overhead due to the need to calculate the advantage function, which can slow down the training. A novel real-time task scheduling method is presented for edge-cloud environments using the A2C algorithm combined with deep reinforcement learning [7]. Proximal policy optimization (PPO) limits the size of policy updates to prevent instability caused by overly large updates [8]. PPO is relatively simple to implement compared to other advanced reinforcement learning algorithms, making it a popular choice in research and practical applications. However, PPO can be sample-inefficient, meaning it often requires a large number of interactions with the environment to achieve good performance, especially in tasks with high-dimensional action spaces. The PPO algorithm is applied to address active collision avoidance for unmanned surface vehicles (USVs) in complex maritime environments. By incorporating a mathematical model of the USV, dynamic obstacle generation, and a reward mechanism, the approach effectively trains a deep convolutional neural network (CNN) through simulations. Deep deterministic policy gradient (DDPG) is particularly well-suited for tasks with continuous action spaces, where traditional discrete-action algorithms like deep-Q network (DQN) struggle [9]. It is an off-policy algorithm that can reuse past experiences stored in a replay buffer. This improves sample efficiency compared to on-policy methods like A2C and PPO. To enhance autonomous drone mobility control, the DDPG-based method is applied to solve real-time obstacle avoidance in challenging environments. The research integrates sensing-aware nonlinear control with human-in-the-loop feedback through human-computer interaction (HCI), enabling the system to adapt to unforeseen scenarios effectively.

III. Reinforcement Learning for Torpedo Guidance with Obstacle Avoidance

Our objective is for the torpedo model to hit a target moving in a straight line within a 2D environment. Fig. 1 plots the modeling of a simplified torpedo environment. The orange curve represents the torpedo's trajectory, while the blue line indicates the target's path. The torpedo starts at (0,0) and aims to avoid obstacles that are randomly generated while pursuing a target moving from the (0,100) direction at a random angle. The torpedo moves at 1.5 times the speed of the target, allowing it to hit the target if it finds an appropriate path. In this system, the state is composed of information about the torpedo, target, and obstacles. The state includes the distance and bearing between the torpedo and the target, as well as the distance and bearing between the torpedo and the obstacles. The distance between the torpedo and an obstacle

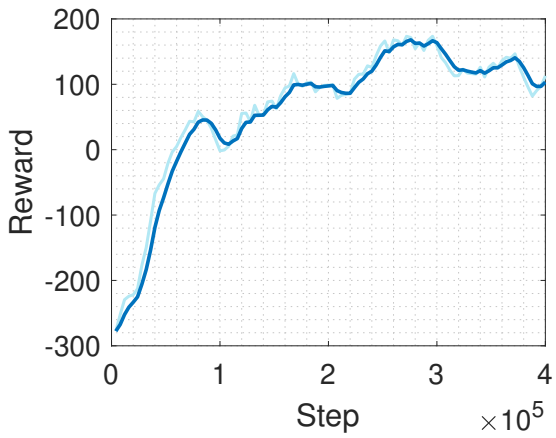

Fig. 2: Reward over training steps

is calculated as the distance between the torpedo and the center of the obstacle, minus the radius of the obstacle. The action includes the torpedo's movement direction, producing a continuous action in degrees ranging from 0 to 359.9. The reward function incorporates factors that guide the torpedo to avoid obstacles and reach its destination. First, as a transition reward, it includes the difference between the distance from the torpedo to the target in the previous step and the current step. Additionally, to prevent collisions with obstacles, a higher reward is given when the absolute value of the bearing between the torpedo and the obstacle is large. If the torpedo collides with an obstacle or fails to hit the target within the specified steps, it receives a large negative reward. Conversely, if it hits the target within the designated steps, it receives a large positive reward, and the episode terminates.

We applied the DARC algorithm, which demonstrates strong performance in highly random environments. It investigates the use of double actors to improve value estimation in reinforcement learning, addressing the overestimation and underestimation biases seen in methods like DDPG [5].

Fig. 1 illustrates the trajectories of the torpedo when the target moves in different directions and obstacles are generated at random positions. It can be observed that the torpedo detects the positions of the obstacles and avoids them to reach the target. Fig. 2 shows the rewards obtained per step as training progresses. It can be seen that the reward increases continuously and converges over time.

IV. Concluding Remarks

In this paper, we explored the application of RL for torpedo guidance with a focus on obstacle avoidance and target acquisition in a dynamic environment. By utilizing the DARC algorithm, we addressed the challenges of overestimation and underestimation biases common in reinforcement learning methods, such as the DDPG. Our approach demonstrated robust performance in highly variable scenarios where obstacles and target positions were randomized, indicating the potential of reinforcement learning in complex and uncertain military applications. Future work could involve expanding the current model to three-dimensional environments and integrating additional environmental factors, such as varying torpedo speeds and more obstacles.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government [MSIT (Ministry of Science and ICT (Information and Communications Technology))] (RS-2024-00439803, SW Star Lab) for Quantum AI Empowered Second-Life Platform Technology. J. Kim is the corresponding author of this paper (joongheon@korea.ac.kr).

References

[1] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, and J.-H. Kim, "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-UAV control," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7086–7096, October 2022.

[2] W. J. Yun, M. Shin, D. Mohaisen, K. Lee, and J. Kim, "Hierarchical deep reinforcement learning-based propofol infusion assistant framework in anesthesia," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 2510–2521, February 2024.

[3] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, November 2019.

[4] H. Lee and S. Park, "Sensing-aware deep reinforcement learning with HCI-based human-in-the-loop feedback for autonomous nonlinear drone mobility control," *IEEE Access*, vol. 12, pp. 1727–1736, January 2024.

[5] J. Lyu, X. Ma, J. Yan, and X. Li, "Efficient continuous control with double actors and regularized critics," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, Vancouber, Canada, June 2022, pp. 7655–7663.

[6] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 48, New York, NY, USA, June 2016, pp. 1928–1937.

[7] J. Lu, J. Yang, S. Li, Y. Li, W. Jiang, J. Dai, and J. Hu, "A2C-DRL: Dynamic scheduling for stochastic edge–cloud environments using A2C and deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 915–16 927, May 2024.

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

[9] 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 *Proc. 4th International Conference on Learning Representations (ICLR)*, Caribe Hilton, San Juan, Puerto Rico, May 2016.