

Indoor Path Planning for an Unmanned Aerial Vehicle via Curriculum Learning

Jongmin Park¹, Sooyoung Jang², and Younghoon Shin^{1*}

¹School of Integrated Technology, Yonsei University,
Incheon, 21983, Korea (jm97, yh.s@yonsei.ac.kr)

²Electronics and Telecommunications Research Institute,
Daejeon, 34129, Korea (sy.jang@etri.re.kr)

* Corresponding author

Abstract: In this study, reinforcement learning was applied to learning two-dimensional path planning including obstacle avoidance by unmanned aerial vehicle (UAV) in an indoor environment. The task assigned to the UAV was to reach the goal position in the shortest amount of time without colliding with any obstacles. Reinforcement learning was performed in a virtual environment created using Gazebo, a virtual environment simulator, to reduce the learning time and cost. Curriculum learning, which consists of two stages was performed for more efficient learning. As a result of learning with two reward models, the maximum goal rates achieved were 71.2% and 88.0%.

Keywords: path planning, curriculum learning, reinforcement learning, unmanned aerial vehicle (UAV)

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) are being studied and used in various fields [1–4]. In the case of a quadcopter [5–8], which is one of the most common types of UAV, its position can be maintained through hovering, which is not possible with a fixed-wing UAV. Various sensors can be mounted on the UAV and the location of the UAV can be determined using global navigation satellite systems (GNSS) [9–13], long-term evolution (LTE) based positioning [14–22], enhanced long-range navigation (eLoran) [23–31], and other techniques [32–34]. UAVs can be used for target searching, weather information acquisition, aerial photography, delivery, communication repeating, and for entertainment using light sources [35–37].

Considering the possibilities of using such UAVs, research is being conducted to optimize the movement of UAVs using artificial intelligence (AI) [38–40]. Specifically, reinforcement learning, which has been widely investigated with the recent development in deep learning, was used in [38]. Reinforcement learning involves learning the optimal behavior in a given situation through actions and rewards and is mainly used in robots and game AI.

In this study, we performed reinforcement learning to ensure that a UAV could reach the goal position in the shortest amount of time while avoiding obstacles. In consideration of learning time and cost, the learning was conducted in a virtual indoor environment created using Gazebo [41], a virtual environment simulator. In addition, for the two reward models that we proposed, curriculum learning was performed to increase the efficiency of learning. First, learning was conducted in an environment without obstacles, then after learning had progressed to a certain level, obstacles were added, and the learning was continued.

Curriculum learning is a machine learning technique that involves learning simple tasks sufficiently and then

progressing to difficult and complex tasks. This technique offers the advantages of generalization and a fast convergence speed [42]. In our study, learning was first performed for a simple path planning to ensure that a UAV could fly quickly to a goal point in an environment without obstacles. After this simple task was learned, the learning was performed in an environment with obstacles to train the UAV to fly to a goal point within a short time while avoiding obstacles. Such curriculum learning is more efficient than learning a difficult task from the beginning.

2. VIRTUAL ENVIRONMENT

To implement the UAV virtual environment and learning environment, Gazebo, Robot Operating System (ROS) [43], and OpenAI Gym [44] were used. Using the building editor in Gazebo, a virtual indoor environment of 30 m × 30 m with a few obstacles was created. Fig. 1(a) shows the virtual indoor environment without obstacles used at the beginning of learning, and Fig. 1(b) shows the a virtual indoor environment with obstacles. Figs. 1(c) and 1(d) present top view images of the environment in Figs. 1(a) and 1(b), respectively. The red squares indicate the coordinate set of the UAV. The UAV randomly selects two coordinates from this set as the starting point and goal point for training.

ROS is a meta-operating system for robots that includes hardware abstraction, low-level device control, implementation of commonly used functionality, message passing between processes, and package management [43]. We used ROS because it can be easily integrated into other robot software frameworks, and many studies on robots or UAVs have utilized ROS. The action of the UAV in the OpenAI Gym learning environment can be transferred to the virtual environment implemented in Gazebo; moreover, information from the sensor mounted on the virtual UAV, such as whether the UAV has col-

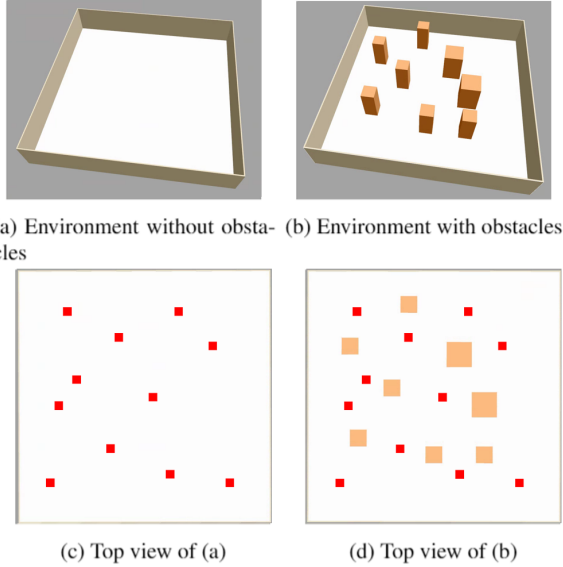


Fig. 1. Virtual indoor environment

lided, can be delivered to the Gazebo learning environment through message passing.

OpenAI Gym is a toolkit for reinforcement learning research [44]. Various learning environments have already been implemented, and new learning environments can be created as per the OpenAI Gym format. In addition, OpenAI Gym is convenient to link with TensorFlow or RLlib [45]. In this study, a learning environment was created based on the OpenAI Gym format, and learning was performed by linking OpenAI Gym with RLlib.

3. LEARNING ENVIRONMENT

The time unit for selecting an action in the current state and receiving a reward is called a *step*, and it is the smallest unit of learning. An *episode* consisting of several steps refers to the time from when the UAV starts the task until it reaches the goal point or collides with an obstacle. When the learning progresses beyond a certain number of episodes, it becomes one *iteration*, and the model parameters of reinforcement learning are updated every iteration.

A *train batch size* of RLlib, which determines the size of one iteration, was set to 10,000 steps in this study, and the learning was started in the environment without obstacles for 200 iterations. The learning was continued in the environment with obstacles for 100 iterations.

The reinforcement learning algorithm used in this study is a proximal policy optimization (PPO) algorithm [46], and is provided with PPOTrainer in RLlib. PPO is a policy gradient-based reinforcement learning method that is more suitable for problems with a continuous state space than for Q-learning-based reinforcement learning such as deep Q-network (DQN) [47–49]. Because the state space in this study was continuous, PPO was selected as the learning algorithm. In this study, learning

rate, named *lr* in RLlib, was set to 5×10^{-5} , trace-decay parameter, named *lambda* in RLlib, was set to 1, and initial coefficient for Kullback-Leibler (KL) divergence, named *kl_coeff* in RLlib, was set to 0.2.

Reinforcement learning is the process of studying the action that maximizes the reward in the current state. Thus, the performance of learning is determined by the state space, action space, and reward model.

3.1 State space

The state space in our study is divided into three types: heading, distance, and lidar data. Heading in this paper refers to the difference in angle between the straight line connecting the UAV and the goal and the heading direction of the current UAV (in radian). Distance refers to the 2D Euclidean distance between the UAV and the goal. Lidar data represents information obtained from a lidar mounted on a UAV.

3.2 Action space

The action space was divided into three forward linear velocities and five yaw rates, and a total of 15 actions were set. The three forward linear velocities were 1 m/s, 0.5 m/s, 0 m/s, and the five yaw rates were $-2/12$ rad/s, $-1/12$ rad/s, 0 rad/s, $1/12$ rad/s, and $2/12$ rad/s. A negative yaw rate indicates turning counterclockwise, while a positive yaw rate indicates turning clockwise. Since the UAV moved in a 2D space, its vertical velocity was set to zero.

3.3 Reward model

Because the reward model is the factor that can have the greatest impact on learning performance, two reward models were designed, and the learning performance was compared between these models. Our reward model is divided into terminal reward, time penalty, progress distance, and progress heading. A difference exists between the two reward models in terms of the progress heading.

Terminal reward is the reward given at the end of the episode. If the task is successful, a reward of +2000 is given, and if the task is failed, a reward of -500 is given. The time penalty is for performing a task within the shortest amount of time, and a reward of -1 is given to each step. Progress distance is a value obtained by multiplying 40 by the difference in the Euclidean distance between the UAV and the goal in the previous step and the current step; it has a positive value when the UAV approaches the goal and a negative value when it moves away from the goal. Progress heading is a reward that varies depending on the heading in the state space; when the absolute value of the heading is less than 20 degrees, a value obtained by multiplying the linear speed by 5 is given as a reward for moving quickly to the goal. In addition, when the absolute value of the heading is greater than 20 degrees, the reward is given by multiplying $\frac{45}{17} \left(\frac{|\text{heading}|}{\pi} - \frac{1}{18} \right)$, which is a linear function of heading, by $-(1 + \text{linear speed})$ for reward model 1 and $-(1 + 3 \times \text{linear speed})$ for reward model 2 to reduce the forward linear velocity and ensure that the UAV heads toward the goal.

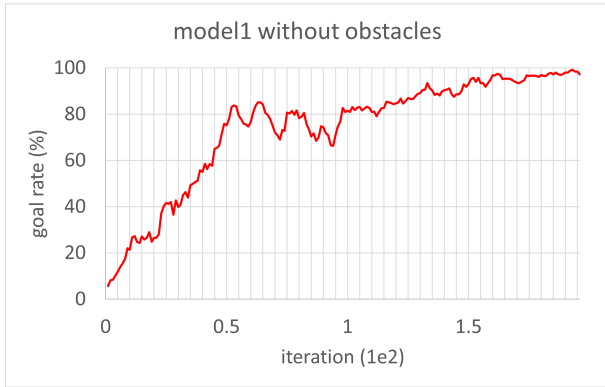


Fig. 2. Moving average of goal rate for reward model 1 in the environment without obstacles

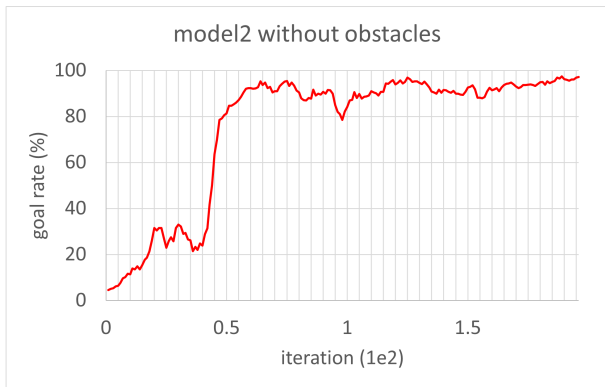


Fig. 3. Moving average of goal rate for reward model 2 in the environment without obstacles

4. SIMULATION RESULTS

4.1 Environment without obstacles

Figs. 2 and 3 show the moving average of goal rate for reward models 1 and 2, respectively, with learning trained for 200 iterations in an environment without obstacles. The moving average was calculated based on the goal rates of the recent five iterations. Reward models 1 and 2 achieved a goal rate of 95.8% and 94.4%, respectively, in the 200 iterations.

4.2 Environment with obstacles

Figs. 4 and 5 show the moving average of goal rate for reward models 1 and 2, respectively, which additionally learned for 100 iterations in an environment with obstacles. Reward models 1 and 2 achieved a maximum goal rate of 71.2% and 88.0%, respectively.

5. CONCLUSION

In this study, we investigated the path planning of a UAV via reinforcement learning, including curriculum learning for two reward models. After learning for 200 iterations in an environment without obstacles, both reward models achieved a high goal rate of approximately

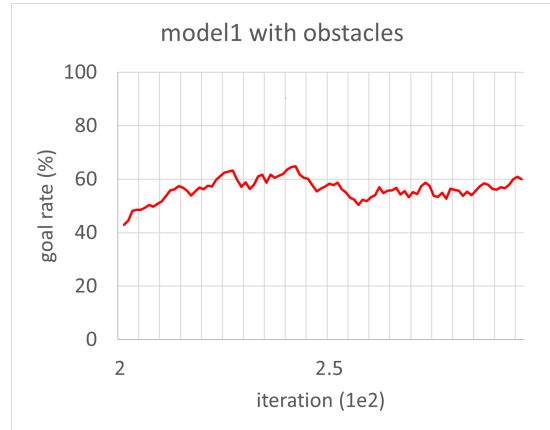


Fig. 4. Moving average of goal rate for reward model 1 in the environment with obstacles

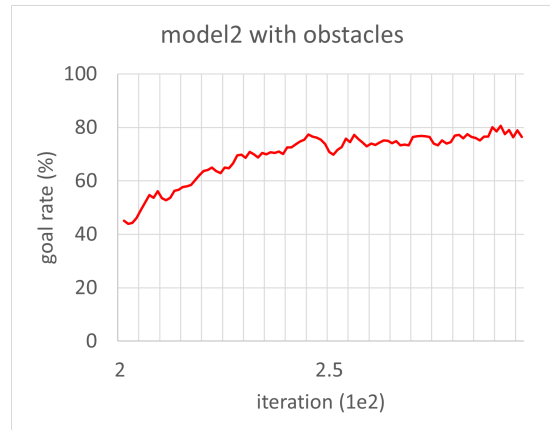


Fig. 5. Moving average of goal rate for reward model 2 in the environment with obstacles

95%. To proceed with curriculum learning, obstacles were added to the environment, and the learning was continued. In the environment with obstacles, the goal rate dropped to approximately 30–40% and then gradually increased again. For reward models 1 and 2, the maximum goal rates were 71.2% and 88%, respectively; thus, reward model 2 outperformed reward model 1. Accordingly, the UAV that learned using reward model 2 reached the goal relatively quickly, without being significantly affected by its initial heading.

ACKNOWLEDGEMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [21ZR1100, A Study of Hyper-Connected Thinking Internet Technology by Autonomous Connecting, Controlling and Evolving Ways].

REFERENCES

- [1] T. Alladi, Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV-UAV communication," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15 068–15 077, 2020.
- [2] H. Lee, T. Kang, and J. Seo, "Development of confidence bound visualization tool for LTE-based UAV surveillance in urban areas," in *Proc. ICCAS*, Oct. 2019, pp. 1187–1191.
- [3] S. K. Singh, K. Agrawal, K. Singh, C.-P. Li, and W.-J. Huang, "On UAV selection and position-based throughput maximization in multi-UAV relaying networks," *IEEE Access*, vol. 8, pp. 144 039–144 050, 2020.
- [4] H. Lee, W. Kim, and J. Seo, "Simulation of UWB radar-based positioning performance for a UAV in an urban area," in *Proc. IEEE ICCE-Asia*, Jun. 2018.
- [5] N. Xuan-Mung, S. K. Hong, N. P. Nguyen, L. N. N. T. Ha, and T.-L. Le, "Autonomous quadcopter precision landing onto a heaving platform: New method and experiment," *IEEE Access*, vol. 8, pp. 167 192–167 202, 2020.
- [6] Y. Shin, S. Lee, and J. Seo, "Autonomous safe landing-area determination for rotorcraft UAVs using multiple IR-UWB radars," *Aerosp. Sci. Technol.*, vol. 69, pp. 617–624, Oct. 2017.
- [7] A. Talaeizadeh, H. N. Pishkenari, and A. Alasty, "Quadcopter fast pure descent maneuver avoiding vortex ring state using yaw-rate control scheme," *IEEE Robot. Autom.*, vol. 6, no. 2, pp. 927–934, 2021.
- [8] J. Kim, J.-W. Kwon, and J. Seo, "Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV," *Electron. Lett.*, vol. 50, no. 20, pp. 1431–1432, Sep. 2014.
- [9] F. Causa and G. Fasano, "Improving navigation in GNSS-challenging environments: Multi-UAS cooperation and generalized dilution of precision," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 3, pp. 1462–1479, 2021.
- [10] K. Sun, H. Chang, J. Lee, J. Seo, Y. Jade Morton, and S. Pullen, "Performance benefit from dual-frequency GNSS-based aviation applications under ionospheric scintillation: A new approach to fading process modeling," in *Proc. ION ITM*, Jan. 2020, pp. 889–899.
- [11] K. Park and J. Seo, "Single-antenna-based GPS antijamming method exploiting polarization diversity," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 2, pp. 919–934, Apr. 2021.
- [12] C. Savas, G. Falco, and F. Dovis, "A comparative performance analysis of GPS L1 C/A, L5 acquisition and tracking stages under polar and equatorial scintillations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 1, pp. 227–244, 2021.
- [13] K. Park, D. Lee, and J. Seo, "Dual-polarized GPS antenna array algorithm to adaptively mitigate a large number of interference signals," *Aerosp. Sci. Technol.*, vol. 78, pp. 387–396, Jul. 2018.
- [14] K. Shamaei and Z. M. Kassas, "A joint TOA and DOA acquisition and tracking approach for positioning with LTE signals," *IEEE Trans. Signal Process.*, vol. 69, pp. 2689–2705, 2021.
- [15] M. Maaref and Z. M. Kassas, "Ground vehicle navigation in GNSS-challenged environments using signals of opportunity and a closed-loop map-matching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2723–2738, 2020.
- [16] T. Kang, H. Lee, and J. Seo, "Analysis of the maximum correlation peak value and RSRQ in LTE signals according to frequency bands and sampling frequencies," in *Proc. ICCAS*, Oct. 2019, pp. 1182–1186.
- [17] S. Jeong, H. Lee, T. Kang, and J. Seo, "RSS-based LTE base station localization using single receiver in environment with unknown path-loss exponent," in *Proc. ICTC*, Oct. 2020, pp. 958–961.
- [18] H. Lee and J. Seo, "A preliminary study of machine-learning-based ranging with LTE channel impulse response in multipath environment," in *Proc. IEEE ICCE-Asia*, Nov. 2020.
- [19] H. Lee, A. Abdallah, J. Park, J. Seo, and Z. Kassas, "Neural network-based ranging with LTE channel impulse response for localization in indoor environments," in *Proc. ICCAS*, Oct. 2020, pp. 939–944.
- [20] H. Lee, J. Seo, and Z. Kassas, "Integrity-based path planning strategy for urban autonomous vehicular navigation using GPS and cellular signals," in *Proc. ION GNSS+*, Sep. 2020, pp. 2347–2357.
- [21] M. Jia, H. Lee, J. Khalife, Z. M. Kassas, and J. Seo, "Ground vehicle navigation integrity monitoring for multi-constellation GNSS fused with cellular signals of opportunity," in *Proc. IEEE ITSC*, 2021.
- [22] T. Kang and J. Seo, "Practical simplified indoor multiwall path-loss model," in *Proc. ICCAS*, Oct. 2020, pp. 774–777.
- [23] P.-W. Son, J. Rhee, J. Hwang, and J. Seo, "Universal kriging for Loran ASF map generation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1828–1842, Oct. 2019.
- [24] P.-W. Son, J. Rhee, and J. Seo, "Novel multichain-based Loran positioning algorithm for resilient navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 2, pp. 666–679, Oct. 2018.
- [25] P. Williams and C. Hargreaves, "UK eLoran—Initial operational capability at the Port of Dover," in *Proc. ION ITM*, 2013, p. 392–402.
- [26] W. Kim, P.-W. Son, J. Rhee, and J. Seo, "Development of record and management software for GPS/Loran measurements," in *Proc. ICCAS*, Oct. 2020, pp. 796–799.
- [27] D. Qiu, D. Boneh, S. Lo, and P. Enge, "Reliable location-based services from radio navigation sys-

- tems,” *Sensors*, vol. 10, no. 12, pp. 11 369–11 389, 2010.
- [28] J. Park, P.-W. Son, W. Kim, J. Rhee, and J. Seo, “Effect of outlier removal from temporal ASF corrections on multichain Loran positioning accuracy,” in *Proc. ICCAS*, Oct. 2020, pp. 824–826.
- [29] J. Hwang, P.-W. Son, and J. Seo, “TDOA-based ASF map generation to increase Loran positioning accuracy in Korea,” in *Proc. IEEE ICCE-Asia*, Jun. 2018.
- [30] Y. Li, Y. Hua, B. Yan, and W. Guo, “Research on the eLoran differential timing method,” *Sensors*, vol. 20, p. 6518, 2020.
- [31] J. H. Rhee, S. Kim, P.-W. Son, and J. Seo, “Enhanced accuracy simulator for a future Korean nationwide eLoran system,” *IEEE Access*, in press.
- [32] E. Kim and J. Seo, “SFOL pulse: A high accuracy DME pulse for alternative aircraft position and navigation,” *Sensors*, vol. 17, no. 10, Sep. 2017.
- [33] J. Rhee and J. Seo, “Low-cost curb detection and localization system using multiple ultrasonic sensors,” *Sensors*, vol. 19, no. 6, Mar. 2019.
- [34] K. Park, W. Kim, and J. Seo, “Effects of initial attitude estimation errors on loosely coupled smartphone GPS/IMU integration system,” in *Proc. IC-CAS*, Oct. 2020, pp. 800–803.
- [35] S. V. Sibanyoni, D. T. Ramotsoela, B. J. Silva, and G. P. Hancke, “A 2-D acoustic source localization system for drones in search and rescue missions,” *IEEE Sensors J.*, vol. 19, no. 1, pp. 332–341, 2019.
- [36] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle routing problems for drone delivery,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, 2017.
- [37] T. Hiraguri, K. Nishimori, I. Shitara, T. Mitsui, T. Shindo, T. Kimura, T. Matsuda, and H. Yoshino, “A cooperative transmission scheme in drone-based networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2905–2914, 2020.
- [38] S. Kim, J. Park, J.-K. Yun, and J. Seo, “Motion planning by reinforcement learning for an unmanned aerial vehicle in virtual open space with static obstacles,” in *Proc. ICCAS*, Oct. 2020, pp. 784–787.
- [39] S. Kouroshnezhad, A. Peiravi, M. S. Haghighi, and A. Jolfaei, “Energy-efficient drone trajectory planning for the localization of 6G-enabled IoT devices,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5202–5210, 2021.
- [40] P. Chhikara, R. Tekchandani, N. Kumar, V. Chamola, and M. Guizani, “DCNN-GA: A deep neural net architecture for navigation of UAV in indoor environment,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4448–4460, 2021.
- [41] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *Proc. IEEE/RSJ IROS*, vol. 3, 2004, pp. 2149–2154.
- [42] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. ICPS*, 2009, p. 41–48.
- [43] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, “ROS: An open-source robot operating system,” in *Proc. ICRA Workshop on Open Source Software*, vol. 3, Jan. 2009.
- [44] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI Gym,” Jun. 2016, arXiv:1606.01540.
- [45] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stolica, “RLlib: Abstractions for distributed reinforcement learning,” in *Proc. ICML*, vol. 80, Jul. 2018, pp. 3053–3062.
- [46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” Jul. 2017, arXiv:1707.06347.
- [47] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013, arXiv:1312.5602.
- [48] S. Y. Jang, H. J. Yoon, N. S. Park, J. K. Yun, and Y. S. Son, “Research trends on deep reinforcement learning,” *Electronics and Telecommunications Trends*, vol. 34, no. 4, pp. 1–14, Aug. 2019.
- [49] M. Klissarov, P.-L. Bacon, J. Harb, and D. Precup, “Learnings options end-to-end for continuous action tasks,” Nov. 2017, arXiv:1712.00004.