OKU Fen Bilimleri Enstitüsü Dergisi
5(2): 783-791, 2022

OKU Journal of The Institute of Science and Technology 5(2): 783-791, 2022

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Evaluation of DDPG and PPO Algorithms for Bipedal Robot Control

Mustafa Can BINGOL[1*]

[1]Fırat University, Faculty of Technology, Department of Mechatronic Engineering, 23160, Elazig, Turkey

[1]https://orcid.org/0000-0001-5448-8281
*Corresponding author: mustafacanbingol@gmail.com

Research Article

Article History:
Received: 03.12.2021
Accepted: 02.03.2022
Published online: 18.07.2022

Keywords:
Bipedal robot
Deep deterministic policy gradient (DDPG)
Proximal policy learning (PPO)
Reinforcement learning (RL)

ABSTRACT

Legged robots are very popular topics in the robotic field owing to walking on hard terrain. In the current study, the walking of a bipedal robot that was a legged robot was aimed. For this purpose, the system was examined and an artificial neural network was designed. After, the neural network was trained by using the Deep Deterministic Policy Gradient (DDPG) and the Proximal Policy Optimization (PPO) algorithms. After the training process, the PPO algorithm was formed better training performance than the DDPG algorithm. Also, the optimal noise standard deviation of the PPO algorithm was investigated. The results were shown that the best results were obtained by using 0.50. The system was tested by utilizing the artificial neural networks that trained the PPO algorithm which has got 0.50 noise standard deviation. The total reward in the test was calculated as 274.334 and the walking task was achieved by purposed structure. As a result, the current study has formed the basis for controlling a bipedal robot and the PPO noise standard deviation selection.

İki Ayaklı Robot Kontrolü için DDPG ve PPO Algoritmalarının Değerlendirilmesi

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 03.12.2021
Kabul tarihi: 02.03.2022
Online Yayınlanma:18.07.2022

Anahtar Kelimeler:
İki Ayaklı Robot
Derin deterministik politika gradyanı
Yakınsal politika öğrenme
Pekiştirmeli öğrenme

ÖZ

Bacaklı robotlar, zorlu arazilerde hareket edebilmeleri nedeniyle robotik alanında çalışılan popüler konulardan biridir. Bu çalışmada, ayaklı bir robot olan iki ayaklı bir robotun yürümesi amaçlanmıştır. Bu amaçla sistem incelenmiş ve bir yapay sinir ağı tasarlanmıştır. Daha sonra Derin Deterministik Politika Gradyanı (Deep Deterministic Policy Gradient - DDPG) ve Yakınsal Politika Optimizasyonu (Proximal Policy Optimization - PPO) algoritmaları kullanılarak sinir ağı eğitilmiştir. Eğitim sürecinden sonra PPO algoritması, DDPG algoritmasına göre daha iyi eğitim performansı oluşturulmuştur. Ayrıca, PPO algoritmasının en iyi gürültü standart sapması araştırılmıştır. Sonuçlar, en iyi sonuçların 0,50 kullanılarak elde edildiğini göstermiştir. Sistem, 0,50 gürültü standart sapmasına sahip PPO algoritmasını eğiten yapay sinir ağları kullanılarak test edilmiştir. Test sonucuna göre toplam ödül 274.334 olarak hesaplanmış ve amaçlanan yapı ile yürüme görevi gerçekleştirilmiştir. Sonuç olarak, mevcut çalışma, iki ayaklı bir robotun kontrol edilmesi ve PPO gürültü standart sapma seçiminin temelini oluşturmuştur.

To Cite: Bingol MC. Evaluation of DDPG and PPO Algorithms for Bipedal Robot Control. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2022; 5(2): 783-791.

1. Introduction

Robots are machines widely used in daily life to facilitate human works from home to industry. Many researchers work in sub-robotic fields such as vision, control, etc. to improve the work missions.

Reinforcement Learning (RL) is one of a lot of developed algorithms to control robots. The main difference of RL to other control algorithms is generated robust control outputs according to dynamic environment effects. RL can be examined two sub-title as value and policy learning. Value learning is a process that chooses the best options from among limited output according to inputs. For example, the artificial structure that can play the Atari game could be made by using value learning. Policy learning is a method that calculates the best continuous output according to inputs. For instance, the controller that controls the positions of the DC motor could be designed by using policy learning. There are a lot of kinds of policy learning such as Deep Deterministic Policy Gradient (DDPG) or Proximal Policy Optimization (PPO).

The DDPG algorithm has been formed by improving the actor-critic method. Some problems have been solved by using the algorithm. For example, 3-DoF planar robot was controlled by an artificial neural network that trained the DDPG algorithm in spite of the damaged actuator (Bingol, 2021a). In another study, inverse kinematic of the 2-DoF planar robot was solved by using the DDPG algorithm and the noise parameter of the DDPG was investigated (Bingol, 2021b). An unmanned surface vehicle has been tracked course by using DDPG (Wang et al., 2018). In other work, solving traffic jam problem via Deep Q Network and DDPG algorithms has been studied (Pang and Gao, 2019). Altitude of quad-copter has been controlled by artificial neural network based on DDPG algorithm (Ghouri et al., 2019).

For instance, mapless navigation problem for mobile robots has been solved by using convolutional PPO (Toan and Woo, 2021). The adaptive metro service schedules problem was solved by utilizing PPO (Ying et al., 2021). In other study, a drone has been controlled by using PPO (Lopes et al., 2018). An unmanned aerial vehicle has been controlled similarly to the previous study (Zhen et al., 2020). A manipulator has tracked planned trajectory via artificial neural network that has been trained by using PPO (Zhang et al., 2019). Inverse kinematic of 5-DoF endoscopic instrument has been solved by using PPO (Schmitz and Berthet-Rayne, 2020).

In the study, it was aimed that a bipedal robot learns to walk. In line with this goal, the bipedal robot environment was analyzed. According to this environment, it was decided that DDPG and PPO algorithms would be suitable learning methods. Actor and critic neural network architectures in DDPG and PPO algorithms were designed. These designed networks were trained and the results were compared.

2. Problem Description

Walking for legged robots is a complex process owing to requiring the robot balance and the unpredictability of the terrain. In this study, the learning of the walking process by a bipedal 4-DoF robot was planned. In accordance with this purpose, firstly a bipedal robot was examined. After, artificial neural networks were formed. Lastly, the neural networks were trained by using DDPG and PPO algorithms.

2.1. Robot Design

A bipedal robot consists of five parts as a hull, two knees, and two hips. These parts could be examined in Figure 1.

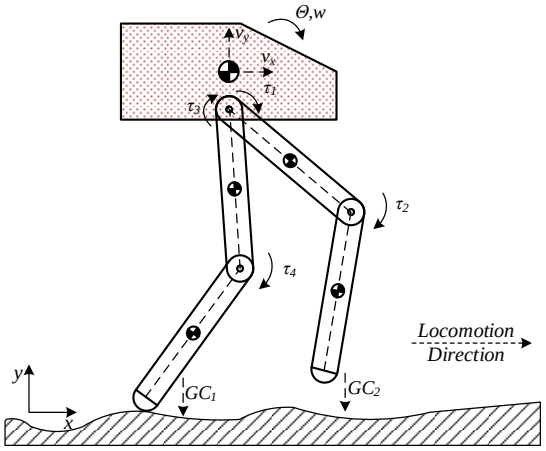


Figure 1. Schematics of mechanical chain a bipedal robot

In Figure 1, $\Theta, w, v, \tau,$ and GC were symbolized as angle of hull, angular velocity of hull, linear velocity of system, motor torque of joint, and ground contact sensors of knee, respectively. The kinematic movements can be modeled by using Euler-Lagrange's motion equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \tau \tag{1}$$

where L represents the Lagrangian, that consist of kinetic $(K(q, \dot{q}))$ and potential energy $(V(q))$ of the system, and could be calculated by using

$$L = K(q, \dot{q}) - V(q). \tag{2}$$

The kinetic energy of system is

$$K(q, \dot{q}) = \frac{1}{2}\dot{q^T} D(q)\dot{q} \tag{3}$$

where $D(q)$ is called as inertia matrix. These equations could be solved as

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) = \tau. \tag{4}$$

In Equation (4), $C(q, \dot{q}), G(q)$, and $F(q, \dot{q})$ were represented as Coriolis, gravity, and friction matrix of system. q and $\dot{q}$ are system parameters such as angle and angular velocity of knees and hips. After the robot locomotion was explained, state vector

$$s = \left[\Theta, w, v_x, v_y, q_1, \dot{q_1}, q_2, \dot{q_2}, q_3, \dot{q_3}, q_4, \dot{q_4}, GC_1, GC_2, l_{1\ to\ 10}\right] \tag{5}$$

was formed. In Equation (5), l was typified as LIDAR sensor output. The sensor was located on the hull in order to perceive the terrain. In the current study, the BipedalWalker-v3 environment (Bipedal-Walker) was used as to simulate the system.

2.2. DDPG Algorithm

DDPG algorithm was formed improving actor-critic method and pseudo code of DDPG algorithm was shown in Algorithm (1).

Algorithm 1. DDPG Algorithm

Initialize CriticNet $Q(s,a|\theta^Q)$ and ActorNet $\mu(s|\theta^\mu)$ with θ^Q and θ^μ weight

Adjust TargetNets weights (Q' and μ') according to θ^Q and θ^μ

Initialize replay buffer (RB) memory

for episode: 1 to 5000

Initialize noise (η)

Reset environment and get s_t

for step: 1 to 500

$a_t = \mu(s_t|\theta^\mu) + \eta_t$

$r_t, s_{t+1}, trm_t = Environment(a_t)$

$(s_t, a_t, r_t, s_{t+1}) \rightarrow RB$

if mod(step, update_coefficent)=0

Get data up to batch size from RB

$y_i = r_i + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$

Update CriticNet using $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$

Update ActorNet using $\nabla_{\theta^\mu} J \simeq \frac{1}{N}\sum_i \nabla_a Q(s_i, \mu(s_i)) \nabla_{\theta^\mu}\mu(s_i|\theta^\mu)$

$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$

$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$

In Algorithm 1, γ and τ were symbolized as discount factor and network update coefficient, respectively. update_coefficent, γ, and τ were chosen as 1, 0.99, and 0.005, respectively. Detailed information about DDPG algorithm was found in (Lillicrap et al., 2016). Also, ActorNet was given in Algorithm (2).

Algorithm 2. Designed artificial neural network structure

$o = \text{Dense Layer}(\text{Unit} = 1024, \text{activation} = \text{ReLU})(s)$
$o = \text{Dense Layer}(\text{Unit} = 1024, \text{activation} = \text{ReLU})(o)$
$o = \text{Dense Layer}(\text{Unit} = 1024, \text{activation} = \text{ReLU})(o)$
$o = \text{Dense Layer}(\text{Unit} = 1024, \text{activation} = \text{ReLU})(o)$
$out = \text{Dense Layer}(\text{Unit} = 4, \text{activation} = \tanh)(o)$

In Algorithm (2), dense layer (He et al., 2015) is where the mathematical operations are done. Also, activation functions were preferred rectified linear unit (ReLU) and hyperbolic tangent (tanh) function. Algorithm 2 is a multilayer perceptron model. The multilayer perceptron model is a feedforward neural network. Each neuron in this network is connected to all neurons in the previous layer.

According to the neurons in the previous layer, it produces the output after the necessary mathematical operations.

2.3. PPO Algorithm

The PPO algorithm has been a popular algorithm among RL algorithms owing to both simple implementation and more effective solutions. The PPO algorithm was given in Algorithm (3).

Algorithm 3. The PPO algorithm for Actor-Critic Method

for episode: 1 to 5000

for actor: 1 to N

Run policy $\pi_{\theta_{\text{old}}}$ in environment for T time steps

Compute advantage estimates $\widehat{A_1}, \ldots, \widehat{A_T}$

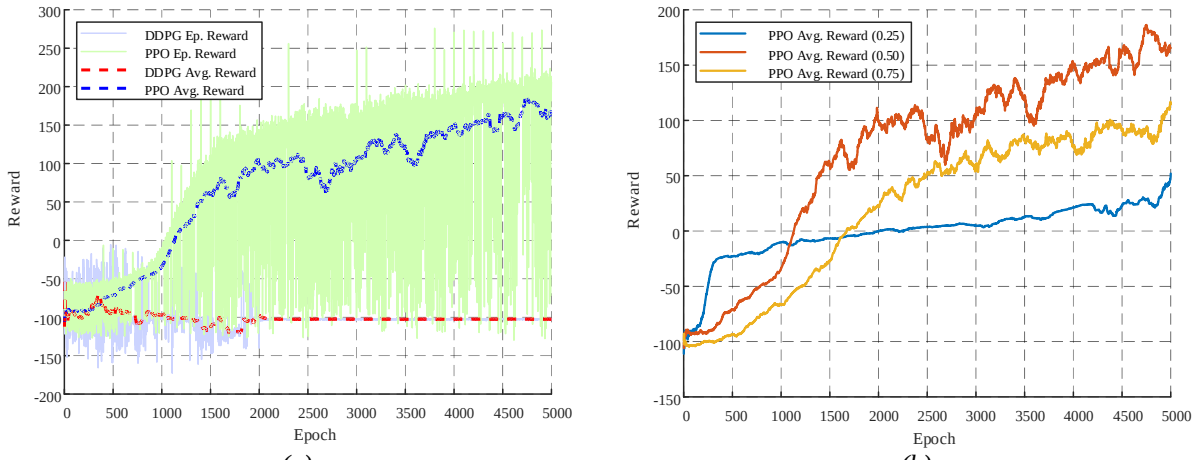
Optimize surrogate L write θ, with K epochs and minibatch size M≤NT

$\theta_{old} \leftarrow \theta$

Algorithm 2 was used as actor network in Algorithm 3. Exploration is one of the natural processes of the reinforcement learning method. In this study, the exploration process was provided by adding noise to the output of the action network. The mean of this noise term was 0 and its standard deviation was determined 0.25, 0.50, and 0.75, respectively. Also, detailed information about PPO was represented in (Schulman et al., 2017).

3. Results and Discussion

In the current study, an actor neural network was designed as in Algorithm 2 and designed actor neural network was trained by DDPG and PPO algorithms which are frequently used deep reinforcement learning methods. Train performances of DDPG and PPO algorithms were shown in Figure 2-a.



(a) *(b)*

Figure 2. Train performance of proposed algorithms; (a) DDPG and PPO, (b) PPOs

Ep. reward and Avg. reward were symbolized epoch reward and average reward in Figure 2. The average reward was calculated by the last 100 epoch reward. Average reward of DDPG algorithm and PPO algorithm were shown with red and blue dotted dash line in Figure 2-a and last reward values were -100 and 150, respectively. Both these values and Figure 2-a showed that the PPO algorithm was better than the DDPG algorithm about controlling the bipedal robot. If the DDPG algorithm is used to a similar problem by practitioners and researchers, the best actor model should be formed and the current actor model is controlled whether it is the best at the end of every epoch. After it was investigated that the PPO algorithm could better control the bipedal robot, the standard deviation of noise was investigated because of the important this hyper-parameter that could be seen in the previous work (Bingol, 2021b). The best result was generated by using 0.50 standard deviation value when Figure 2-b was analyzed. This study was carried out with the help of OpenAI Gym using Python language on Google Colab. The used PC specifications were Intel Xeon 2.20GHz CPU, Nvidia Tesla P100-16GB GPU, 12GB RAM, and 160GB Disk memory. The training span of PPO algorithms that was standard deviation 0.25, 0.50, and 0.75 were 238, 222, and 214 minutes by using the PC, respectively. The best PPO algorithm that was standard deviation 0.50 was retrained and the training span was measured as 287 minutes. After these training process was done, the robot was tested as in Figure 3.

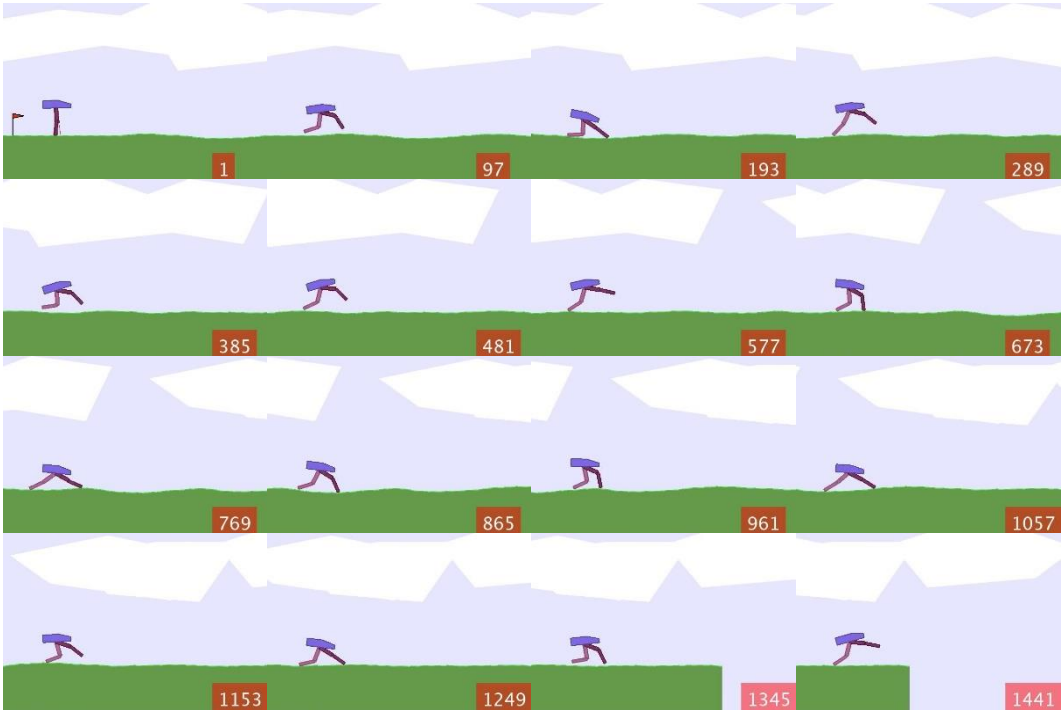
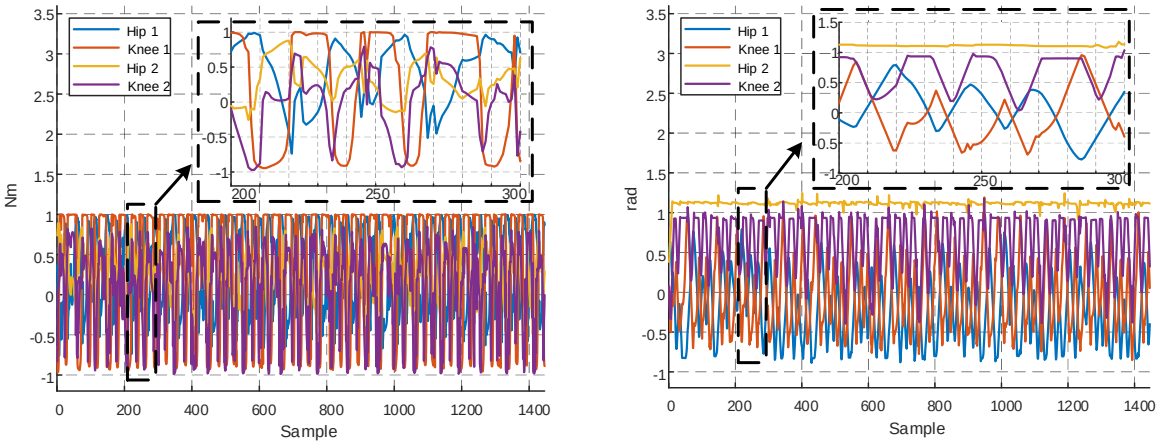


Figure 3. Test of the bipedal robot control

In Figure 3, sample moment of test was located at right bottom of each picture. The bipedal robot achieved the walking task and total reward was calculated as 274.334.



(a) *(b)*

Figure 4. System values during the test; (a) Motors torques, (b)Joints angles

During the task, applied motors torques and measured joints angles were given in Figure 4.

4. Conclusion

In the current study, controlling of the bipedal robot was aimed. Firstly, the bipedal robot was examined and then an artificial neural network was designed. The neural network was trained by using DDPG and PPO algorithms. The PPO algorithm was better than the DDPG algorithm about controlling locomotion of then bipedal robot according to training results, which can be seen in Figure 2-a. The environment used in the study is not frequently used in the literature. Therefore, studies are limited. BipedalWalker environment was used in a study (Dong et al., 2021). When this study was examined, the average reward was calculated as -100 after 50000 steps. In our study, the average reward after 5000 steps was measured as roughly 150. Then, standard deviation of noise that is in PPO algorithm was investigated. According to the result of this investigation, the standard deviation of the noise is a very important hyper-parameter to train the PPO algorithm, the same way as DDPG (Bingol, 2021b). The best result was obtained using 0.50 standard deviation value, could be seen in Figure 2-b. A very important point of Figure 2-b is the transient response of the system. The best result at the first 500 epochs was measured by using a 0.25 standard deviation value whereas the worst result was obtained in a long term. The reason for the situation is that noise is less than dynamic effects. The early-stage of the training process could be misled at the PPO algorithm owing to the situation. Lastly, high noise standard deviation (0.75) extends training time, as can be seen in Figure 2-b. The best method for determining the noise standard deviation in dynamic systems is to know the system. Training time changes depending on done situation at RL system. Thus, the best PPO structure was trained twice times and training spans were obtained as 222 and 287 minutes.

In the future works, advanced algorithms that will reduce training time and increase performance will be tested on the same system.

Statement of Conflict of Interest

Author has declared no conflict of interest.

Author's Contributions

The contribution of the author's is 100%.

References

Bingol MC. Development of neural network based on deep reinforcement learning to compensate for damaged actuator of a planar robot. Global Conference on Engineering Research (GLOBCER'21) June 2021a; 310–317.

Bingol MC. Investigation of the standard deviation of ornstein - Uhlenbeck noise in the DDPG Algorithm. Gazi University Journal of Science Part C: Design and Technology 2021b; 9(2): 200–210.

Dong Y., Zhang S., Liu X., Zhang Y., Shen T. Variance aware reward smoothing for deep reinforcement learning. Neurocomputing 2021; 458, 327–335.

Ghouri UH., Zafar MU., Bari S., Khan H., Khan MU. Attitude control of quad-copter using deterministic policy gradient algorithms (DPGA). 2019 2nd International Conference on Communication, Computing and Digital Systems C-CODE 2019; 149–153.

He K., Zhang X., Ren S., Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision 2015; 1026-1034.

https://github.com/openai/gym/blob/b6b4fc38388c42c76516c93fd107dab124af64dd/gym/envs/box2d/bipedal_walker.py (Accessed October 14, 2021) Bipedal-Walker

Lillicrap TP., Hunt JJ., Pritzel A., Heess N., Erez T., Tassa Y., Wierstra D. Continuous control with deep reinforcement learning. 4th International Conference on Learning Representations, ICLR 2016.

Lopes GC., Ferreira M., Da Silva Simoes A., Colombini EL. Intelligent control of a quadrotor with proximal policy optimization reinforcement learning. 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education, 2018; 509–514.

Pang H., Gao W. Deep Deterministic policy gradient for traffic signal control of single intersection. 31st Chinese Control and Decision Conference 2019; 5861–5866.

Schmitz A., Berthet-Rayne P. Using Deep-learning proximal policy optimization to solve the inverse kinematics of endoscopic instruments. IEEE Transactions on Medical Robotics and Bionics 2020; 3(1): 273–276

Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. Proximal Policy optimization algorithms. 2017.

Toan ND., Woo KG. Mapless navigation with deep reinforcement learning based on the convolutional proximal policy optimization network. 2021 IEEE International Conference on Big Data and Smart Computing 2021; 298–301.

Wang Y., Tong J., Song TY., Wan ZH. Unmanned surface vehicle course tracking control based on neural network and deep deterministic policy gradient algorithm. 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans Oceans - 2018; 3–7.

Ying CS., Chow AHF., Wang YH., Chin KS. Adaptive metro service schedule and train composition with a proximal policy optimization approach based on deep reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 2021; 1–12.

Zhang S., Pang Y., Hu G. Trajectory-tracking control of robotic system via proximal policy optimization. 9th International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics 2019; 380–385.

Zhen Y., Hao M., Sun W. Deep reinforcement learning attitude control of fixed-wing UAVs. 3rd International Conference on Unmanned Systems 2020; 239–244.