SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ

Sakarya University Journal of Science
SAUJS

e-ISSN 2147-835X Period Bimonthly Founded 1997 Publisher Sakarya University
http://www.saujs.sakarya.edu.tr/

Title: Reinforcement Learning-Based Safe Path Planning for a 3R Planar Robot

Authors: Mustafa Can BİNGOL

Recieved: 2021-04-08 00:00:00

Accepted: 2021-12-28 00:00:00

Article Type: Research Article

Volume: 26
Issue: 1
Month: February
Year: 2022
Pages: 128-135

How to cite
Mustafa Can BİNGOL; (2022), Reinforcement Learning-Based Safe Path Planning for a 3R Planar Robot. Sakarya University Journal of Science, 26(1), 128-135, DOI: 10.16984/saufenbilder.911942
Access link
http://www.saujs.sakarya.edu.tr/tr/pub/issue/67934/911942

Sakarya University Journal of Science 26(1), 128-135, 2022

Reinforcement Learning-Based Safe Path Planning for a 3R Planar Robot

Mustafa Can BİNGOL*[1]

Abstract

Path planning is an essential topic of robotics studies. Robotic researchers have suggested some methods such as particle swarm optimization, A*, and reinforcement learning (RL) to obtain a path. In the current study, it was aimed to generate RL-based safe path planning for a 3R planar robot. For this purpose, firstly, the environment was performed. Later, state, action, reward, and terminate functions were determined. Lastly, actor and critic artificial neural networks (ANN), which are basic components of deep deterministic policy gradients (DDPG), were formed in order to generate a safe path. Another aim of the current study was to obtain an optimum actor ANN. Different ANN structures that have 2, 4, and 8-layers and 512, 1024, 2048, and 4096-units were formed to get an optimum actor ANN. These formed ANN structures were trained during 5000 episodes and 200 steps and the best results were obtained by 4-layer, 1024, and 2048-units structures. Owing to this reason, 4 different ANN structures were performed utilizing 4-layer, 1024, and 2048-units. The proposed structures were trained. The NET-M2U-4L structure generated the best result among 4 different proposed structures. The NET-M2U-4L structure was tested by using 1000 different scenarios. As a result of the tests, the rate of generating a safe path was calculated as 93.80% and the rate of colliding to the obstacle was computed as 1.70%. As a consequence, a safe path was planned and an optimum actor ANN was obtained for a 3R planar robot.

Keywords: Artificial neural networks, Deep Deterministic Policy Gradients, path planning, reinforcement learning

1. INTRODUCTION

Path planning is a popular topic in field of robot. Therefore, many researchers have studied this topic and many methods have been developed to solve this task. To illustrate, an optimum path for mobile robots has been generated by using improvement adaptive ant colony algorithm [1]. In another study, a smooth path has been planned utilizing particle swarm optimization and high-degree Bezier curve [2]. In other study, path planning process has been realized faster by using bidirectional associate learning [3]. Path planning process for an industrial robot has been implemented by using genetic algorithm [4]. An improved A* algorithm has been compared to original A* algorithm and the developed algorithm has generated path which is short and high success rate [5]. Another path planning algorithm is reinforcement learning (RL). For example, a path for reconfigurable robot platform has been planned by using RL [6].

* Corresponding author: mustafacanbingol@gmail.com
[1] Fırat University, Faculty of Technology, Department of Mechatronics
ORCID: https://orcid.org/0000-0001-5448-8281

RL when compare to other learning methods such as supervised or unsupervised is more suitable to control robot motion because the method needn't a dataset. Many studies have been carried out utilizing RL in the literature [6]–[12]. For example, Matulis and Harvay have realized a digital twin of an articulated robot using P-proximal policy optimization (PPO) that is a RL method [8]. In another study, MIMO PID controller for mobile robots has been tuned using based on deep deterministic policy gradients (DDPG) method which is a RL method [12]. In other study, hybrid and end-to-end DDPG structures have been formed to control cable driven parallel robot. Two structures have controlled the robot but learn hybrid learned faster than end-to-end structure in the study [7]. The deep p-network and dueling deep p-network have been developed for reaching task of a n-DoF robot. Then, the structures have been realized filliping of a handkerchief and folding a t-shirt tasks [11]. In another study, energy-efficient and adaptive control structures for snake-like robot have been developed by using RL and inverse RL algorithms [10]. A robot has been navigated in mazes by utilizing RL [9].

In the current study, a safe path for 3R planar robot was planned by using DDPG. Also, to obtain an optimum actor artificial neural network (ANN), the proposed 12 different actor ANN structures were designed. The ANN structures, that generated the best results, were mixed and an optimum actor ANN structure was obtained.

2. MATERIALS AND METHODS

The developed system consists of two parts as environment and agent. The environment part contains the 3R planar robot kinematics, obstacle, and target positions. There are ANN that plan safe paths in the agent part. The detailed information about these parts and the communication between these parts were shown in Figure 1.

The main blocks and sub-blocks was shown in Figure 1. State (s), action (a), reward (r), and terminate (trm) variables were used the same as traditional RL algorithms to communicate parts with each other and train networks. The state contains 17 data which include six distances (d_i) between sensors and obstacle, six angles (γ_i) between sensors and obstacle, angles of three joints (θ_i), one distance (d) between the tool center point (TCP) and target, and one angle (γ) between TCP and target. The state variables were shown in Figure 2.

Sensors, target, and obstacle was illustrated orange, green, and red filled circles, respectively. d, γ, d_i, and γ_i were calculated by using Equations (1-4).

$$d = \sqrt{(x_{TCP} - x_{TRG})^2 + (y_{TCP} - y_{TRG})^2} \quad (1)$$

$$\gamma = tan^{-1}((x_{TCP} - x_{TRG}), (y_{TCP} - y_{TRG})) \quad (2)$$

$$d_i = \sqrt{(x_{SNS_i} - x_{OBS})^2 + (y_{SNS_i} - y_{OBS})^2} \quad (3)$$

$$\gamma_i = tan^{-1}((x_{SNS_i} - x_{OBS}), (y_{SNS_i} - y_{OBS})) \quad (4)$$

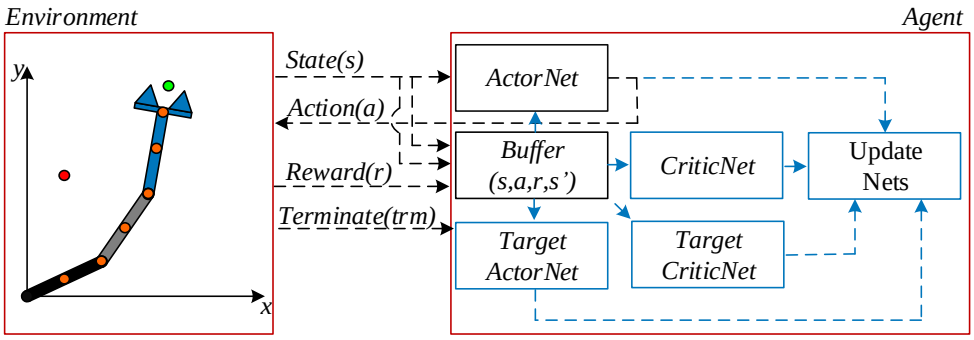


Figure 1 Block diagram of developed system

In Equations (1-4), (x_{TCP}, y_{TCP}), (x_{TRG}, y_{TRG}), and (x_{SNS_i}, y_{SNS_i}) were typified as position of TCP, target and sensor, respectively. After these calculations, s was obtained as [d, γ, d_1, γ_1, …, d_6, γ_6, θ_1, θ_2, θ_3]. Also, in the system, length of robot links was chosen 0.1 m. Position of target was randomly determined according to $x_{TRG} = 0.05 + 0.15\Omega\cos(\pi\Psi)$ and $y_{TRG} = 0.05 + 0.15\Omega\sin(\pi\Psi)$ equations. Here, Ω and Ψ were randomly generated numbers between 0 and 1. Position of obstacle was randomly determined according to $x_{OBS} = x_{TRG} + 0.05 + 0.05\Omega\cos(\pi\Psi)$ and $y_{OBS} = y_{TRG} + 0.05 + 0.05\Omega\sin(\pi\Psi)$ equations. Thus, when the TCP of the robot moves to the target, the possibility of collision that is between the robot and the obstacle increases with the obstacle equations. The TCP of the 3R robot could be calculated by using Table 1 and Equations (5-6).

Table 1 The 3-DoF Planar Robot DH Parameters

Link (i)	a_i	α_i	d_i	θ_i
1	0	0	l_1	$\boldsymbol{\theta_1}$
2	0	0	l_2	$\boldsymbol{\theta_2}$
3	0	0	l_3	$\boldsymbol{\theta_3}$

$$T_i^{i-1} = R_{Z_{i-1}}(\theta_i)T_{Z_{i-1}}(d_i)T_{X_i}(a_i)R_{X_i}(\alpha_i) \quad (5)$$

$$T_3^0 = T_1^0 T_2^1 T_3^2 \quad (6)$$

The kinematic model of the robot was formed by using the homogeneous transformation matrix (T_i^{i-1}). The R_{axis} and T_{axis} were typfied as revlution and translation relevant axis, respectively. Also, link lengths were represented as l in Table 1. The action provides control of the robot. In the current study, to control the robot, the action was designed three parameters as Δθ1, Δθ2, and Δθ3. The $\Delta\theta_i$ parameter was symbolized angle change between two times. The range of $\Delta\theta_i$ was chosen as ±3°. As a result, input of developed system a was designed as [Δθ1, Δθ2, Δθ3].

One of the important topics in RL is the reward function. The reward function directly affects learning performance. The designed reward function was given in Algorithm 1.

In Algorithm 1, thr was symbolized obstacle threshold distance, was chosen 0.025m, and could be seen in Figure 2. Also, the collision refers to whether the robot and the obstacle collided.

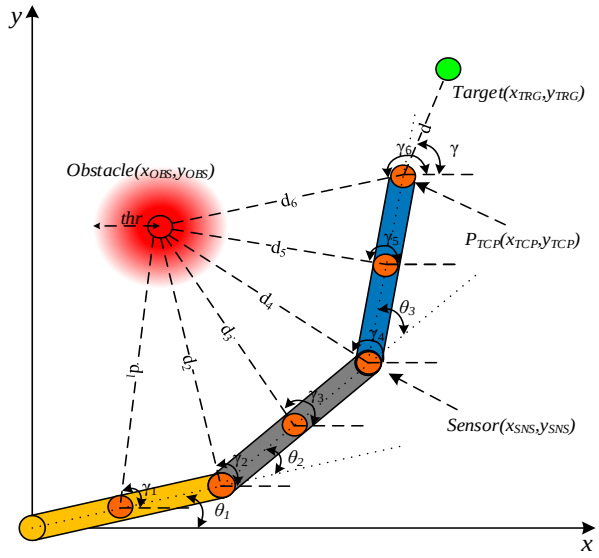


Figure 2 The 3R planar robot

Algorithm 1. Reward Function

function r(d, d_1, d_2, d_3, d_4, d_5, d_6, thr, collision)
$out = -10d$
for i: 1 to 6
  if d_i< thr
    $out = out + \log(d_i)$
if collision=1
  $out = -100$
if d<0.001
  $out = 100$
return out

Terminate function is that the desired operating has occurred or the system has broken down. For example, the state of a game over or arrived finish could be a terminate function for a car game. Terminate function was given in Equation (7) for the current study.

$$trm = \begin{cases} 1, & d < 1mm \text{ or } collision = 1 \\ 0, & other\ conditions \end{cases} \quad (7)$$

In Equation (7), d<0.001 condition was expressed that desired operating occurred and collision=1 condition was shown that the system was damaged. Hence, the system was terminated at these two conditions. The modeled system was simulated in Open AI Gym environment.

The developed 3R planar robot environment was controlled by utilizing DDPG Algorithm. The DDPG Algorithm was given in Algorithm 2. In

order to get detailed information about Algorithm 2, [13] could be examined.

Algorithm 2 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 200
    $a_t = \mu(s_t|\theta^\mu) + \eta_t$
    $r_t, s_{t+1}, trm_t = Environment(a_t)$
    $(s_t, a_t, r_t, s_{t+1}) \rightarrow RB$
    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 2, update_coefficent, discount factor (γ), and network update coefficient (τ) were chosen as 1, 0.99, and 0.01, respectively. In this study, there was no static dataset since the online RL algorithm was applied. The replay buffer could be called a dataset, and the replay buffer consists of the current state, action, reward, and next-step state data. In the current study, 10^6 was selected as replay buffer size and batch size was chosen as 64. Also, there are 4 ANN structure (2 ActorNets and 2 CriticNets), in Algorithm 2. These ANN's were trained by using replay buffer. In the current study, it was aimed to obtain optimum actor ANN for safe path planning. The actor ANN was given in Algorithm 3.

Algorithm 3 ActorNet Algorithm

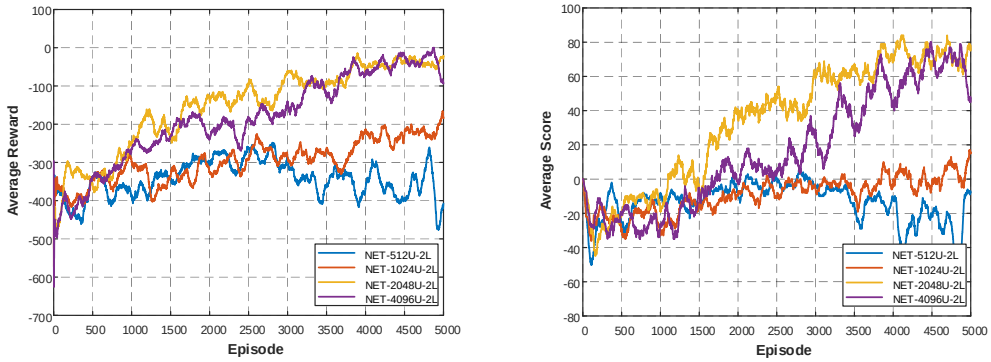
function ActorNet(U,L)
  $o =$ Dense Layer (Unit = U, activation = ReLU)(s)
for step: 1 to L-1
  $o =$ Dense Layer (Unit = U, activation = ReLU)(o)
  $out =$ Dense Layer (Unit = 3, activation = tanh)(o)
return out

In Algorithm 3, dense layer [14] was preferred as layer type. U and L was shown that unit and layer count, respectively. In the current study, U was chosen 512, 1024, 2048, and 4096. L was determined as 2, 4, and 8. Thus, 12 different actor ANN was formed. The actor ANN models were named as NET-XXXXU-YL. Here, X and Y refers unit and layer count. After this step, the models were trained. As a result of the training, the best results were obtained from 4 layers 1024 and 2048-units networks. Because of this reason, 4 different actor ANN were performed as additionally. Unit count of layer in these 4 different actor ANN were 1024-2048-2048-1024 (NET-M1U-4L), 2048-1024-1024-2048 (NET-M2U-4L), 1024-2048-1024-2048 (NET-M3U-4L), and 2048-1024-2048-1024 (NET-M3U-4L).

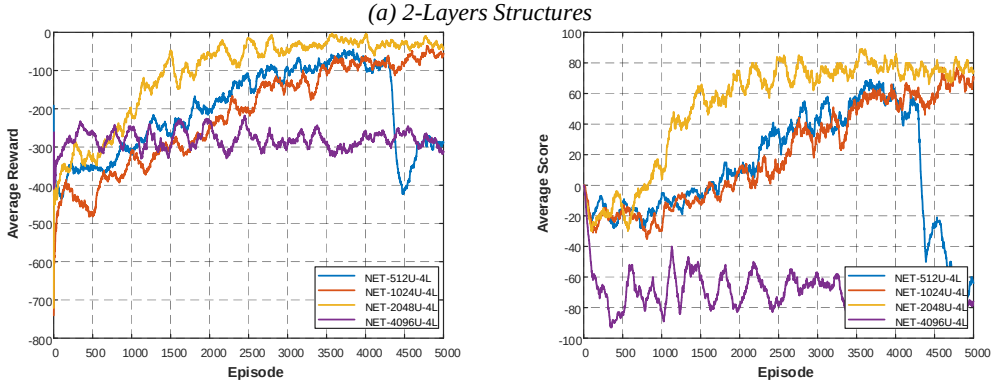
3. RESULTS

The proposed 12 actor ANN structures were trained to plan safe path for the 3R planar robot. Each train process contains 5000 episodes and each episode consists of 200 steps. Also, Adam optimizer was utilized and learning rates were chosen as 10^{-4}. When training actor ANN structures, obtained average reward and score graphs were illustrated in Figure 3.

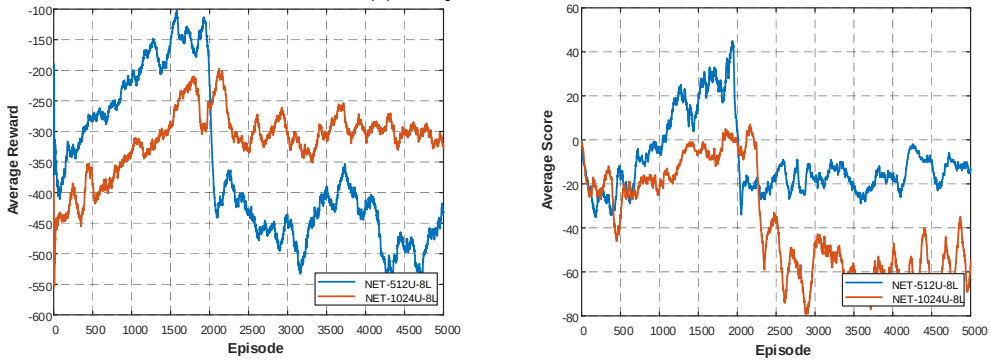
In Figure 3, average reward and score were calculated averaging last 100 episode rewards and scores, respectively. Score was calculated subtracting from number of target arrived by TCP to number of collision occurred. Results of NET-2048U-8L and NET-4096U-8L couldn't be shown because unstable results were obtained. When examined in Fig. 3-b, the best results were obtained by using 1024U,2048U-4L networks. Not only the reward but also the score of these networks were the best around the proposed networks. Descriptive analysis of the training rewards was given in Table 2.



(a) 2-Layers Structures



(b) 4-Layers Structures



(c) 8-Layers Structures

Figure 3 Training process graphs of ANN structures

Table 2 Descriptive analysis of the training rewards

Layer Count (n)	$\bar{X} \pm SD$	Min	Max
2 Layer (n: 20000)	-244.906 ±117.484	-625.914	0.488
4 Layer (n: 20000)	-203.830 ±121.018	-740.419	-2.500
8 Layer (n: 10000)	-336.793 ±96.777	-553.043	-102.201

In Table 2, mean of average rewards ($\bar{X}$), standard deviation (SD), maximum, and minimum values were given according to layer counts. Also, these layer groups were compared with each other utilizing Tukey multiple comparison method. The result of comparison between groups was given in Table 3.

Table 3 Group comparison of layer counts

Group (I)	Group (J)	p
2 Layer	**4 Layer**	<0.001
	8 Layer	<0.001
4 Layer	**2 Layer**	<0.001
	8 Layer	<0.001
8 Layer	**2 Layer**	<0.001
	4 Layer	<0.001

The three groups were found different from each other, as can be seen in Table 3. When Figure 3 and Table 2-3 were examined, 4-layer structure was most successful. The most successful results were obtained in 1024 and 2048 units among the 4-layer structures. In the line with this result, NET-M1U-4L, NET-M2U-4L, NET-M3U-4L, and NET-M4U-4L ANN structures, which were

mentioned in Section 2, were designed. The designed ANN structures were trained, as can be seen in Figure 4. After training process of proposed ANN structures, the rewards results were analyzed and the analysis results were given in Table 4.

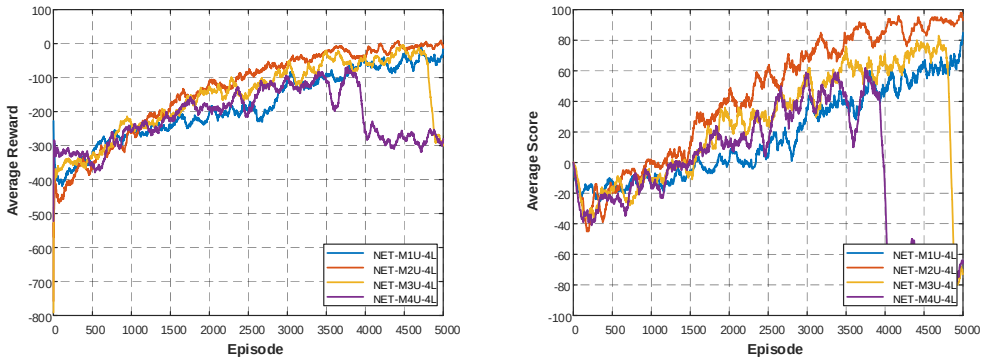


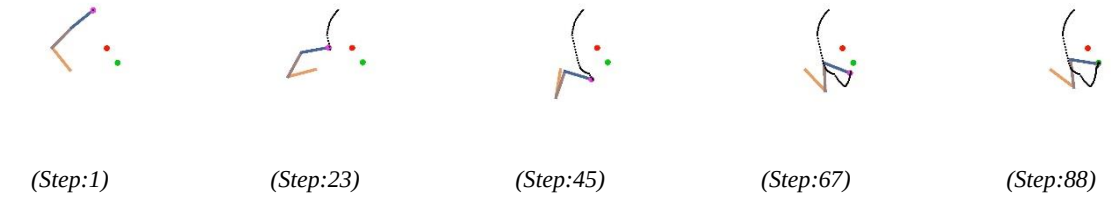
Figure 4 Training process graphs of proposed ANN structures

When Figure 4 and Table 4 were figured out, the best result was obtained from the NET-M2U-4L structure. Therefore, after the training process, the NET-M2U-4L structure was tested as in Figure 5.

The proposed structure successfully passed the single tests, as can be seen in Figure 5. Moreover, video result of these test and more was given in https://youtu.be/sB3wPZMt_pk. After single tests, the structure was tested for 1000 different scenarios. In the test scenarios, the rate of reaching target of the TCP was calculated as 93.80%. The rate of occurring collision was obtained as 1.70% in these scenarios. In other conditions, the median of the distance between TCP and target were calculated as 2.7mm (maximum:0.155m and minimum:0.001m). The current study was carried out according to targets and obstacles which are in located I and II regions according to the Cartesian coordinate system. The proposed algorithm will be generated solutions according to the targets and obstacles that are in located other regions if the system can be trained for a long time.

Table 4 Descriptive analysis of the proposed ANN structures training rewards

NET (n)	$\bar{X} \pm SD$	Min	Max
NET-M1U-4L (n: 5000)	-180.993 ±100.771	-419.801	-8.658
NET-M2U-4L (n: 5000)	-134.144 ±129.831	-757.891	9.330
NET-M3U-4L (n: 5000)	-157.434 ±106.348	-791.622	-3.334
NET-M4U-4L (n: 5000)	-218.359 ±77.787	-522.497	-67.575



(Step:1) *(Step:23)* *(Step:45)* *(Step:67)* *(Step:88)*

(a) Test 1

(Step:1) *(Step:15)* *(Step:29)* *(Step:43)* *(Step:58)*

(b) Test 2

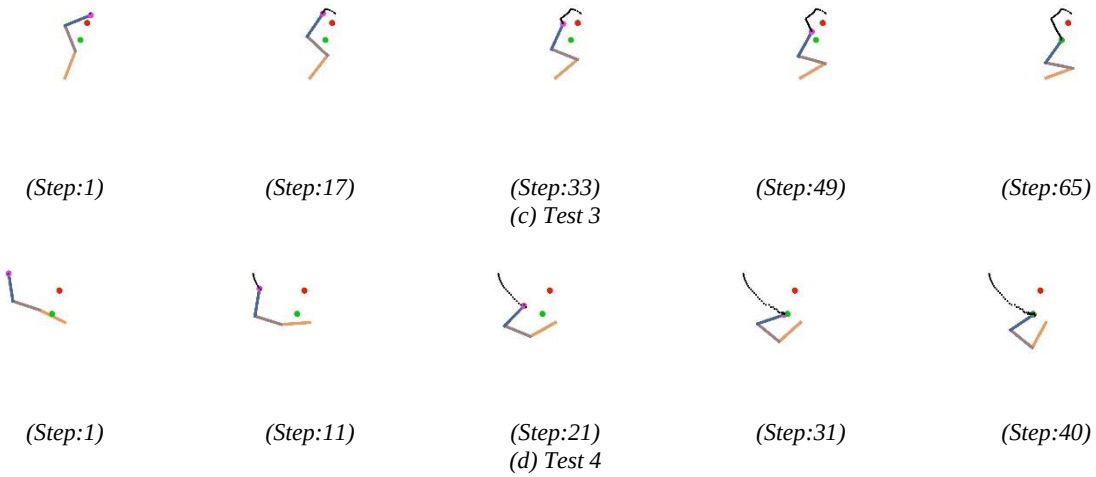


Figure 5 Test of NET-M2U-4L structure

4. CONCLUSIONS

Path planning is an important problem for robotic researchers. Some methods have been developed in order to solve the important problem and one of these developed methods is RL. In the current study, it was aimed that RL-based safe path planning for the 3R planar robot. In the line with this aim, firstly, the 3R planar robot environment was designed. When the environment was designed, state, action, reward, and terminate functions, which are basic components of RL, were determined. After, DDPG Algorithm that is a policy learning method was chosen to control the robot. DDPG Algorithm consists of actor and critic ANN structures. Another aim of the current study was that obtain optimum actor ANN. For this purpose, different ANN structures that have 2, 4, and 8-layers and 512, 1024, 2048, and 4096-units, were trained. After the training process, the best results were obtained from 4-layers ANN structures. Especially among 4-layers ANN structures, the best results were obtained in 1024 and 2048-units. Therefore, 4 different ANN structures, that have mixed unit count, were proposed. The best result was obtained by the proposed NET-M2U-4L structure among these structures. Later, the NET-M2U-4L structure was tested for 1000 different scenarios. In these scenarios, the rate of reaching the target of the TCP was calculated as 93.80%. Also, the rate of occurring collision was obtained as 1.70%. As a result, the RL-based safe path was significantly planned by the developed system and optimum actor ANN was obtained. In future works, the performance of system will be improved and the developed system will be implemented on more advanced robots such as industrial robots.

Funding

The author has no received any financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

Authors' Contribution

The study was performed by one author.

The Declaration of Ethics Committee Approval

The study doesn’t need any ethics committee approval or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data

collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] Y. Zheng, Q. Luo, H. Wang, C. Wang, and X. Chen, "Path planning of mobile robot based on adaptive ant colony algorithm," J. Intell. Fuzzy Syst., vol. 39, no. 4, pp. 5329–5338, 2020.

[2] B. Song, Z. Wang, and L. Zou, "An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve," Appl. Soft Comput., vol. 100, 2021.

[3] M. Zhao, H. Lu, S. Yang, Y. Guo, and F. Guo, "A fast robot path planning algorithm based on bidirectional associative learning," Comput. Ind. Eng., vol. 155, no. October 2020, p. 107173, 2021.

[4] L. Larsen and J. Kim, "Path planning of cooperating industrial robots using evolutionary algorithms," Robot. Comput. Integr. Manuf., vol. 67, no. August 2020, 2021.

[5] B. Fu et al., "An improved A* algorithm for the industrial robot path planning with high success rate and short length," Rob. Auton. Syst., vol. 106, pp. 26–37, 2018.

[6] A. Krishna Lakshmanan et al., "Complete coverage path planning using reinforcement learning for Tetromino based cleaning and maintenance robot," Autom. Constr., vol. 112, no. January, 2020.

[7] H. Xiong, T. Ma, L. Zhang, and X. Diao, "Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots," Neurocomputing, vol. 377, pp. 73–84, 2020.

[8] M. Matulis and C. Harvey, "A robot arm digital twin utilising reinforcement learning," Comput. Graph., vol. 95, pp. 106–114, 2021.

[9] J. Wang, S. Elfwing, and E. Uchibe, "Modular deep reinforcement learning from reward and punishment for robot navigation," Neural Networks, vol. 135, pp. 115–126, 2021.

[10] Z. Bing, C. Lemke, L. Cheng, K. Huang, and A. Knoll, "Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning," Neural Networks, vol. 129, pp. 323–333, 2020.

[11] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," Rob. Auton. Syst., vol. 112, pp. 72–83, 2019.

[12] I. Carlucho, M. De Paula, and G. G. Acosta, "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," ISA Trans., vol. 102, pp. 280–294, 2020.

[13] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 2016.

[14] "Dense Layer." [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense. [Accessed: 27-Mar-2021].