Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 26, S. 358-363, Temmuz 2021
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 26, pp. 358-363, July 2021
Copyright © 2021 EJOSAT
Research Article

Deep Reinforcement Learning Based Controller Design for Model of The Vertical Take-off and Landing System

Mahmut Ağralı[1*], Mehmet Uğur Soydemir[1], Alkım Gökçen[1], Savaş Şahin[1]

[1] Izmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Electrical and Electronics Engineering, İzmir, Türkiye
mahmutagrali4209@gmail.com, soydemirmehmetugur@gmail.com, alkim.gokcen@outlook.com, sahin.savas@yahoo.com
(ORCID: 0000-0002-5508-2854, 0000-0002-2327-1642, 0000-0002-8131-388X, 0000-0003-2065-6907)

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(**DOI:** 10.31590/ejosat.957216)

ATIF/REFERENCE: Ağralı, M., Soydemir, M. U., Gökçen, A. & Şahin, S. (2021). Deep Reinforcement Learning Based Controller Design for Model of The Vertical Take-off and Landing System. *European Journal of Science and Technology*, (26), 358-363.

Abstract

In this study, the Deep Deterministic Policy Gradient (DDPG) algorithm, which consists of a combination of artificial neural networks and reinforcement learning, was applied to the Vertical Takeoff and Landing (VTOL) system model in order to control the pitch angle. This algorithm was selected because conventional control algorithms such as Proportional-Integral-Derivative (PID) controllers which cannot always generate a suitable control signal eliminating the disturbance and unwanted environment effects on the considered system. In order to control the system, training was carried out for a sinusoidal reference in the mathematical model of the VTOL system in the Simulink environment, through the DDPG algorithm with continuous action space from deep reinforcement learning methods that can produce control action values that take the structure that can maximize the reward according to a determined reward function for the purpose of control and the generalization ability of artificial neural networks. For sinusoidal reference and a constant reference, tracking error performances obtained for the pitch angle, which is the output for the specified VTOL system, were compared with the conventional PID controller performance in terms of mean square error, integral square error, integral absolute error, percentage overshoot and settling time. The obtained results are presented via the simulations studies.

Keywords: Reinforcement Learning, DDPG, PID, VTOL.

Dikey Kalkış ve İniş Sistemi Modeli için Derin Pekiştirmeli Öğrenme Tabanlı Kontrolör Tasarımı

Öz

Bu çalışmada, yapay sinir ağları ve pekiştirmeli öğrenmenin birleşiminden oluşan Deep Deterministic Policy Gradient (DDPG) derin pekiştirme öğrenme algoritması Dikey Kalkış ve İniş (VTOL) sistemi modeline yunuslama (pitch) açısını kontrol edebilme amacıyla uygulanmıştır. Bu algoritma, Oransal-İntegral-Türevsel (PID) kontrolör gibi geleneksel kontrol algoritmaları için en uygun kontrolör katsayıları bulunsa dahi kontrol edilecek sistem üzerindeki bozucu etki ve istenmeyen ortam etkilerini elimine edebilecek kontrol sinyali üretememelerinden dolayı seçilmiştir. Belirtilen bu problemi çözebilmek için kontrol amacına yönelik belirlenen bir ödül fonksiyonuna göre ödülü maksimize edebilecek yapısı ve yapay sinir ağlarının genelleştirme yeteneğini arkasına alan kontrol aksiyon değerleri üretebilen derin pekiştirmeli öğrenme yöntemlerinden sürekli eylem uzayına sahip DDPG algoritmasının, Simulink ortamında VTOL sisteminin matematiksel modelinde sinüzoidal bir referans için eğitimi gerçekleştirilmiştir. Belirtilen VTOL sistemi için çıkış olan yunuslama açısının, DDPG algoritması için sinusoidal ve sabit referans için elde edilen izleme başarımları, geleneksel PID kontrolör algoritmasının izleme başarımları ile ortalama kare hatası, integral kare hatası, integral mutlak hatası, yüzde aşım ve oturma zamanı cinsinden karşılaştırılmıştır ve edinilen sonuçlar simülasyon çalışmaları ile sunulmuştur.

Anahtar Kelimeler: Pekiştirmeli Öğrenme, DDPG, PID, VTOL.

[*] Corresponding Author: İzmir Kâtip Çelebi Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, İzmir, Türkiye, ORCID: 0000-0002-5508-2854, mahmutagrali4209@gmail.com

1. Introduction

Deep reinforcement learning (DRL) based algorithms are commonly used methods in the field of controller system design due to their generalization ability, and performance against the possible disturbance effects (Lillicrap et al., 2015). DRL based controller design has a significant role on flow, speed, temperature, position, and process control applications (Rabault et al., 2019; Chen et al., 2018; Brandi et al., 2020; Satheeshbabu et al., 2019; Spielberg et al., 2017). The DRL algorithms consider the environment and agent action pairs to compute a control signal which maximize a pre-determined reward function over a policy (Sutton and Barto, 2018; Buşoniu et al., 2018). For the reinforcement learning based controllers, a Q learning based adaptive method, which considers model states, is employed to compute optimal Proportional-Integral-Derivative (PID) controller gains for a nonlinear cart-pole plant (Shi et al., 2018). Rahman et al. employed both Q learning and deep Q learning (DQN) based controller methods to control a self-balancing robot model, compared the results considering the reference trackings, and evaluated the reward scaling factor selection effects on cumulative reward (Rahman et al., 2018). Markov decision process (MDP) based Fitted Value Iteration (FVI) controller method, which considers quadratic state terms, is employed for an altitude control process of unmanned aerial vehicles (UAVs) (Bou-Ammar et al., 2010). Qin et al. experimented with the reverse pendulum system by determining the constant parameters of conventional control methods such as PID through the DDPG-based reinforced learning algorithm (Qin et al., 2018). Hu et al. have designed DDPG-based controller to control its pressure of Variable Geometry turbocharger system and compared the performances of the conventional PID controller and the designed controller (Hu et al., 2019). Hossny et al. compared the performance of the proposed method using a parameterized tanh activation function instead of the normal tanh activation function in the artificial neural network to the performance of the unparameterized tanh activation function on DDPG algorithm in bipedal walk, lunar lander and reverse pendulum problems by testing (Hossny et al., 2020). Parvaresh et al. proposed a controller to control the pitch angle of the variable-speed wind turbine by using a DDPG based nonlinear integral backstepping algorithm and tested the controller on the different scenarios (Parvaresh et al., 2020).

In the study, Deep Deterministic Policy Gradient (DDPG) controller algorithm is employed for tracking control of the vertical take-off and landing (VTOL) system model pitch angle. The system model is implemented on MATLAB/Simulink environment. DDPG-based controller parameters and working conditions are determined for the simulation process. DDPG agent is trained in each randomly initialized episode for a sinusoidal reference signal. Control process is repeated for sinusoidal and constant reference signals. Mean-squared-error (MSE), Integral-Squared-Error (ISE), Integral-Absolute-Error (IAE) and time measures of transient parts are computed to analyze both reference tracking performance and closed-loop system dynamics. The conventional PID controller algorithm is employed to evaluate the performance of the proposed DDPG based controller algorithm.

The rest of the article is as follows: In the Section 2, the mathematical background of the model of the VTOL system, model parameters and the DDPG algorithm are explained. Section 3 presents the implementation conditions, hyperparameter selection, and simulation scenario details. Herein, performance evaluation metrics of the simulations are explained, and simulation results for both PID and proposed method-based controller system are presented. Conclusions and possible future directions of the study are given in Section 4.

2. Material and Method

In this part of the study, the mathematical model and parameters of the VTOL system and control algorithm are explained.

2.1. VTOL System Model

The VTOL mechanism shown as a free body diagram in Fig. 1 is a useful tool used to show the basics of aircraft such as quadcopters, helicopters and rockets. Flight dynamics can be examined using it and vertical take off and landing control can be provided. The VTOL system consists of a weight that can be relocated and a dc motor fan that can change speed (Junejo et al., 2020; Quanser, 2011). The transfer function of the VTOL system is shown in Eq.1.

$$\frac{Y(s)}{U(s)} = \frac{3.11}{s^2 + 0.576s + 10.7} \tag{1}$$

where Y(s) refers to the Θ angle variable which is the output of the system and U(s) refers to the voltage variable that is the input of the system. The parameters, values, and units considered in the extraction of the transfer function of the VTOL system are shown in Table 1 (Quanser, 2011).

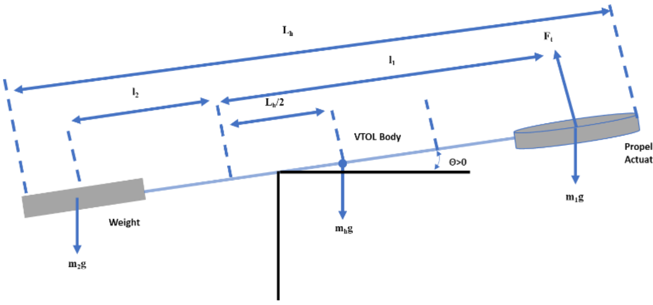


Fig. 1. The Free Body Diagram of the VTOL System

Table 1. The VTOL System Parameters

Parameter	Symbol	Value	Unit
Equilibrium Current	$\mathrm{I_{eq}}$	1.0	A
Torque-thrust Constant	$\mathrm{K_t}$	0.0226	$(Nm)/A$
Moment of Inertia	J	0.0035	$kgm2$
Viscous Damping	B	0.002	$(Nms)/rad$
Natural Frequency	$\mathrm{\omega_n}$	2.52	rad
Stiffness	K	0.022	$(Nm)/rad$
Measured Torque-thrust Constant	$\mathrm{K_{tid}}$	0.01	$(Nm)/A$
Measured Viscous Damping	$\mathrm{B_{id}}$	0.006	$(Nms)/rad$
Measured Stiffness	$\mathrm{K_{id}}$	0.015	$(Nm)/rad$
Length of the setup	$\mathrm{L_h}$	0.3	m

2.2. Deep Deterministic Policy Gradient

DDPG is a model-free, off-policy, actor-critic type deep reinforcement learning algorithm (Lillicrap et al., 2015). Model-free structure directly provides to use experiences which are obtained in an environment ε without needing to find estimates of them while off-policy nature means that estimate of optimal policy is different from behavior policy which is used to choose actions. Actor-critic type can be considered as value-based and policy-based, so this type of algorithm uses both a value function and a policy function. DDPG algorithm includes 4 network which are $Q(s,a|\theta^Q)$ function as a network (critic), $\mu(s|\theta^\mu)$ is a deterministic policy function (actor) which denotes current policy and it directly provides a mapping from observations to action, Q' is target Q network and μ' is target μ network shown in Fig. 3. Considering the Eq. 2, $Q(s',a')$ is dependent to Q function while Q is being updated according to Eq. 2 (Bellman equation)

$$Q_{new}(s,a) = Q(s,a) + \alpha(R(s,a) + \gamma \max Q(s',a') - Q(s,a)) \qquad (2)$$

so, this will make divergence problem, where, $Q_{new}(s,a)$ is new Q function, $Q(s,a)$ is the current Q function, $\max Q(s',a')$ is the maximum expected future reward, $R(s,a)$ is the reward which is taken after applying to action in state s, α is the learning rate, γ is the discount rate. This problem is solved by using target networks which are delayed copies of actual networks in terms of time. Algorithm is given as follows: critic $Q(s,a|\theta^Q)$ network with weights θ^Q, actor $\mu(s|\theta^\mu)$ network with weights θ^μ are initialized. Then, target networks Q' and μ' are initialized with same weights in their actual counterparts. This target Q' and μ' networks are related to training stability. Their predicted values are used in the Bellman equation instead of $Q(s',a')$ to train (update) main Q network (value function). As a technique known experience replay is constructed as experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ tuple at a time t step where s_t are observations (states) at time t, a_t is action at time t, r_t is reward at time t, s_{t+1} are observations at time $t+1$ are stored in $R = e_1, e_2, \dots, e_N$ dataset which is known as replay memory. Replay memory (R) and capacity N of replay memory are initialized. In each episode, a random process N is initialized for exploration purpose so unlike discrete action space (algorithms in discrete action space realize exploration using Boltzman distribution or epsilon-greedy algorithm), exploration is provided to adding a noise (N) to policy function $\mu(s|\theta^\mu)$ for continuous action space. Generally, random process N is Ornstein-Uhlenbeck Process. After initializing random process N, initial observations are obtained. For each time step t, action at t is selected according to $a_t = \mu(s|\theta^\mu) + \mathrm{N}$. Then, action is applied to environment in s_t, observations s_{t+1}, and reward r_t are obtained. Then, experience $e_t = (s_t, a_t, r_t, s_{t+1})$ is stored in *replay memory* R. Next, a random mini batch is extracted from R (so data correlations are reduced) and size of this random mini batch M is defined by user. For every ith experience in minibatch, $target_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{u'})|\theta^{Q'})$ is calculated, loss function $L = \frac{1}{M}\sum_{i=1}^{M}(target_i - Q(s_i,a_i)|\theta^Q)^2$ is minimized with respect to θ^Q parameters and $Q(s,a|\theta^Q)$ is updated. Then, $\nabla_{\theta^\mu} J \approx \frac{1}{M}\sum_{i=1}^{M}\nabla_a Q(s,a|\theta^Q)|s = s_i, a = \mu(s_i)\nabla_{\theta^\mu}\mu(s|\theta^\mu)|s_i$ sampled policy gradient is used for updating the policy $\mu(s|\theta^\mu)$. Main networks parameters θ^Q, θ^μ and target networks parameters $\theta^{Q'}$and $\theta^{\mu'}$ are synchronized (equality of parameters of both network) for every C steps (periodically) which is a parameter defined by user or target network parameters can be updated using one of another target update methods (Lillicrap et al., 2015*;* Sutton and Barto, 2018). When the reinforcement learning algorithm is implemented in control systems, a policy, and an environment corresponds the controller, all things excluding the controller, respectively (Fig. 2). Observation(s), action, reward denote measured variable(s) (system states, output tracking error), control signal, a function which is for control purpose(s) (tracking error dependent function).

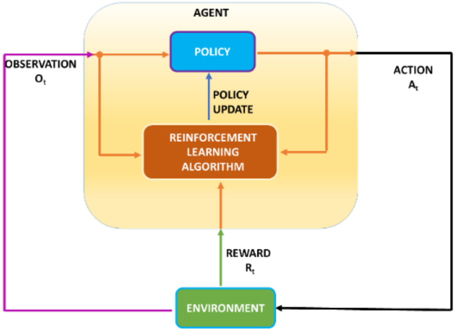
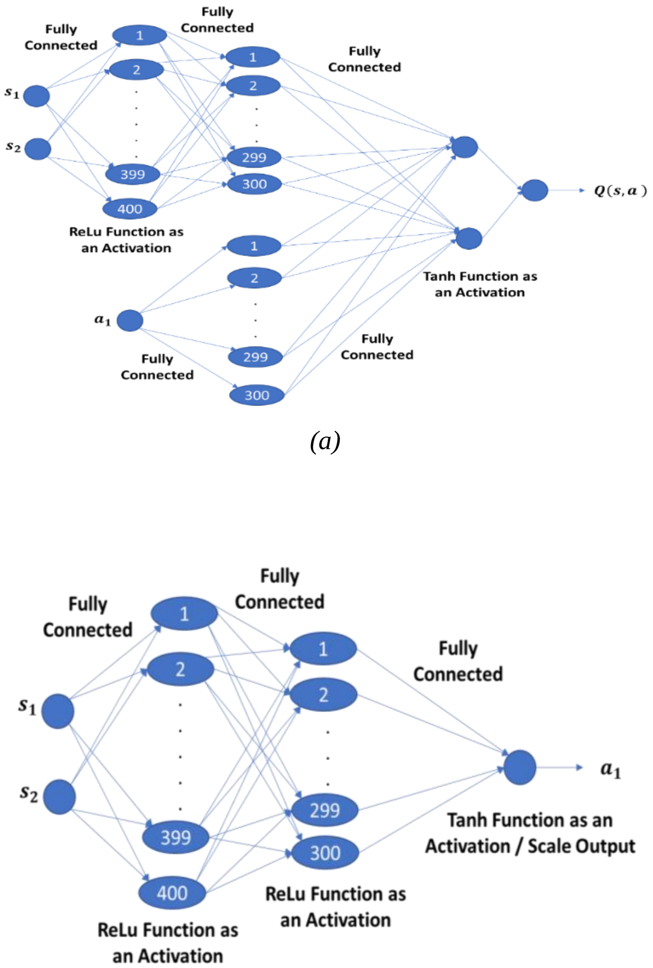


Fig. 2. The reinforcement learning controller structure for a system in an environment

3. Results and Discussion

The DDPG as control algorithm was implemented at MATLAB/Simulink simulation software environment by using the Reinforcement Learning Toolbox and The Deep Learning Toolbox. The controller algorithms and mathematical model of VTOL system are run with personel computer having Intel Core i7-8750 CPU 2.2GHz microprocessor, GeForce RTX 2070 as GPU, 32 GB of RAM, and Windows 10 operating system.

The observations for DDPG were selected as tracking error of VTOL's pitch angle, time derivative of pitch angle and the reward function was determined as $r(t) = -10e^2(t)$, where $e(t)$ is tracking error of pitch angle so best possible reward is 0 according to reward function. Continuous action space was chosen as interval [-6, 6]. Critic network and actor network structures were created as in Fig.3. The determined hyper parameters of the DDPG based control algorithm are given in the Table 2.



(a)

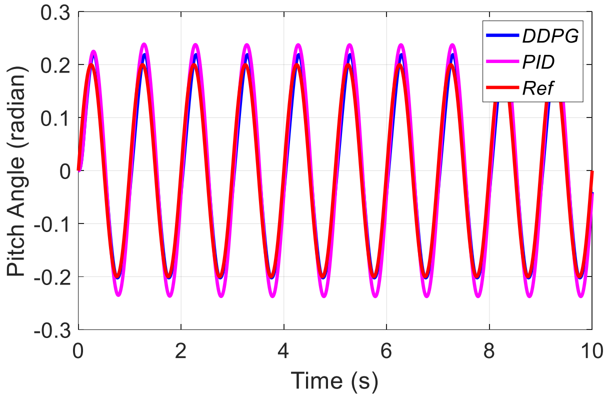
(b)

Fig. 3. The critic network structure $Q(s, a)$ (a) and actor network structure $\mu(s)$ (b)

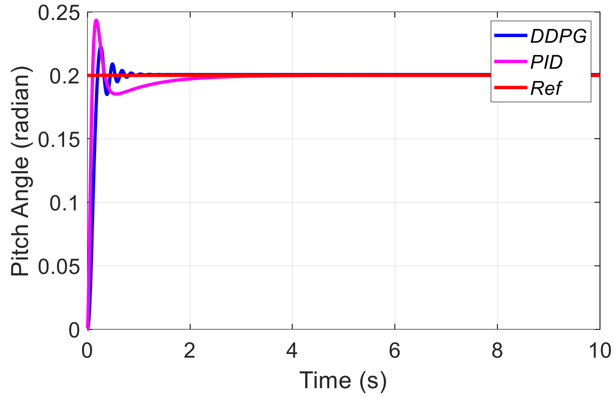
Table 2. The Hyper Parameters of the DDPG algorithm

Hyper Parameters	***Value of Hyper Parameters***
Learning rate (for the critic network $Q(s, a\|\theta^Q)$)	*1e-3*
Learning rate (for the actor network $\mu(s\|\theta^\mu)$)	*1e-04*
Gradient threshold	*1*
DDPG Agent sample time in terms of seconds	*0.01*
Experience buffer length (N)	*1e6*
Discount factor (γ)	*0.99*
Mini batch size (M)	*128*
Training device	*Geforce RTX 2070 as gpu*

14 hours after training of DDPG algorithm for sinusoidal reference $0.2sin(2\pi)$ was started, the most suitable agent for the control purpose has been observed in the end of 701st episode which has a reward -11. For this reinforcement agent, the 10 seconds responses of the sinusoidal and constant signals for the desired pitch angle of the VTOL system by using the PID and DDPG based control algorithm were shown in the Fig.4a and Fig.4b, respectively. The PID controller parameters are tuned by MATLAB PID tuner application as $K_p = 29.599$, $K_i = 34.108$ and $K_d = 4.607$ (Taşören et al., 2020). According to result of sinusoidal signal as a desired output that is shown in Table 3, the performance of DDPG based control algorithm is better than the PID based control algorithm in terms of MSE, ISE and IAE. However, the performance of PID based control algorithm that is given in Table 4 is better than the performances of the DDPG based control algorithm for constant signal as a desired output in terms of MSE and ISE but DDPG constant reference performance in terms of settling time according to %2 criterion and percentage overshoot are smaller than PID one as given in Table 5 (Ogata, 2010).



(a)



(b)

Fig. 4. The Tracking Performance of the PID and DDPG Based Controller for Desired Outputs (a) Sinusoidal and (b) Constant Reference

Table 3. The results of the control algorithms in terms of MSE, ISE and IAE for sinusoidal reference

	MSE	ISE	IAE
DDPG	7.6089×10^{-4}	0.008054	0.2451
PID	0.0014	**0.01493**	**0.3481**

Table 4. The results of the control algorithms in terms of MSE, ISE and IAE for constant reference

	MSE	ISE	IAE
DDPG	5.7666×10^{-4}	0.003369	0.03137
PID	4.0157×10^{-4}	0.00187	0.03223

Table 5. The results of the control algorithms in terms of settling time and overshoot for constant reference

	Settling Time (s)	Overshoot (%)
DDPG	0.79	11.24
PID	2.29	21.69

4. Conclusions and Recommendations

In this study, the DDPG based control algorithm is implemented to control the pitch angle of the VTOL system model through MATLAB/Simulink environment. The DDPG based control algorithm are tested for the pitch angle in terms of sinusoidal and constant signals as desired outputs. The obtained results are compared to the PID based control algorithm whose parameters are tuned by Simulink PID tuner application, in terms of MSE, ISE, IAE, settling time and percentage overshoot. The tracking error performance of DDPG based control algorithm for a sinusoidal reference is better than the PID based control algorithm in terms of MSE, ISE, IAE. The tracking error performance of the DDPG based control algorithm for constant reference is not as good as the PID control algorithm in terms of all error metrics, but it is better in terms of percentage overshoot and settling time than PID control algorithm for constant reference response. so DDPG as a controller can be used in fast and sensitive systems. In the future studies, the other reinforcement-based algorithms that have continuous action space can be used to control the VTOL system model.

5. Acknowledge

This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under 2209A – Research Project Support Programme for Undergraduate Students with project number 1919B012002772.

References

Bou-Ammar, H., Voos, H., & Ertel, W. (2010, September). Controller design for quadrotor uavs using reinforcement learning. In 2010 IEEE International Conference on Control Applications (pp. 2130-2135). IEEE.

Brandi, S., Piscitelli, M. S., Martellacci, M., & Capozzoli, A. (2020). Deep Reinforcement Learning to optimise indoor temperature control and heating energy consumption in buildings. Energy and Buildings, 224, 110225.

Buşoniu, L., Bruin, T., Tolić, D., Kober, J., & Palunko, I. (2018). Reinforcement learning for control: Performance, stability, and deep approximators. Annual Reviews in Control, 46, 8-28.

Chen, P., He, Z., Chen, C., & Xu, J. (2018). Control strategy of speed servo systems based on deep reinforcement learning. Algorithms, 11(5), 65.

Hossny, M., Iskander, J., Attia, M., & Saleh, K. (2020). Refined continuous control of ddpg actors via parametrised activation. arXiv preprint arXiv:2006.02818.

Hu, B., Yang, J., Li, J., Li, S., & Bai, H. (2019). Intelligent control strategy for transient response of a variable geometry turbocharger system based on deep reinforcement learning. Processes, 7(9), 601.

Junejo, M., Kalhoro, A. N., & Kumari, A. (2020). Fuzzy logic based PID auto tuning method of QNET 2.0 VTOL.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Ogata, K. (2010). Modern control engineering. Prentice hall.

Parvaresh, A., Abrazeh, S., Mohseni, S. R., Zeitouni, M. J., Gheisarnejad, M., & Khooban, M. H. (2020). A Novel Deep Learning Backstepping Controller-Based Digital Twins

Technology for Pitch Angle Control of Variable Speed Wind Turbine. Designs, 4(2), 15.

Qin, Y., Zhang, W., Shi, J., & Liu, J. (2018, August). Improve PID controller through reinforcement learning. In 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC) (pp. 1-6). IEEE.

Quanser Inc. (2011) QNET VTOL Instructor Workbook, ftp://ftp.ni.com/evaluation/academic/ekits/QNET_VTOL_Workbook_Student.pdf.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U., & Cerardi, N. (2019). Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. Journal of fluid mechanics, 865, 281-302.

Rahman, M. M., Rashid, S. H., & Hossain, M. M. (2018). Implementation of Q learning and deep Q network for controlling a self balancing robot model. Robotics and biomimetics, 5(1), 1-6.

Satheeshbabu, S., Uppalapati, N. K., Chowdhary, G., & Krishnan, G. (2019, May). Open loop position control of soft continuum arm using deep reinforcement learning. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 5133-5139). IEEE.

Shi, Q., Lam, H. K., Xiao, B., & Tsai, S. H. (2018). Adaptive PID controller based on Q-learning algorithm. CAAI Transactions on Intelligence Technology, 3(4), 235-244.

Spielberg, S. P. K., Gopaluni, R. B., & Loewen, P. D. (2017, May). Deep reinforcement learning approaches for process control. In 2017 6th international symposium on advanced control of industrial processes (AdCONIP) (pp. 201-206). IEEE.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Taşören, A. E., Gökçen, A., Soydemir, M. U., Şahin, S. (2020). Artificial Neural Network-Based Adaptive PID Controller Design for Vertical Takeoff and Landing Model. European Journal of Science and Technology, (Special Issue), 87-93.