2021 7th International Conference on Condition Monitoring of Machinery in Non-Stationary Operations (CMMNO) | 978-1-6654-3386-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CMMNO53328.2021.9467649

Speed tracking of Brushless DC motor based on deep reinforcement learning and PID

Puwei Lu
Mechanical & Electrical Engineering College,
Guangzhou University-Guangzhou-
510000, PR China
2111907042@e.gzhu.edu.cn

Wenkai Huang *
Mechanical & Electrical Engineering College,
Guangzhou University-Guangzhou-
510000, PR China
16796796@qq.com

Junlong Xiao
Mechanical & Electrical Engineering College,
Guangzhou University-Guangzhou-
510000, PR China
1707200071@e.gzhu.edu.cn

***Abstract*—In order to improve the speed tracking accuracy of Brushless DC motor, this paper proposes a PID control method based on deep deterministic policy gradient algorithm(DDPG) compensation. The running state of BLDCM is monitored by DDPG, and the proportional, integral and differential links of PID are compensated to make PID controller have learning ability. At the same time, the Gauss function and the maximum critical speed interval are combined to define the reward function to enhance the versatility of the control strategy. Simulation results show that the proposed strategy has better speed tracking accuracy than PID controller.**

Keywords- DC motor; speed tracking; DDPG; PID; Gauss function.

I. INTRODUCTION

Brushless DC motor has the advantages of fast response, easy adjustment and stable performance, which is widely used in robots, electric vehicles, household appliances, medical devices, aerospace and various industrial fields [1]. Because brushless DC motor is a multivariable nonlinear system with strong coupling and complex dynamic model, it is a very important challenge to improve the speed tracking accuracy of Brushless DC motor.

Due to the traditional control method of Brushless DC motor control effect is not ideal. Therefore, in recent years, many experts and scholars at home and abroad put forward a variety of advanced control methods to solve the problem. Heri et al. [2] proposed an adaptive neuro fuzzy inference system for speed control of Brushless DC motor. Compared with conventional PID and fuzzy PID controller, it has better performance, but it depends on expert knowledge rule base. Zheng et al. [3] proposed a nonlinear sliding mode controller, because the chattering problem of sliding mode control itself is difficult to overcome, which affects the overall performance of the controller. Hu et al. [4] Proposed a fuzzy PID controller based on Genetic Algorithm. However, genetic algorithm is stochastic and easy to fall into local optimal solution.

In a large number of research and industrial applications, PID controller with its simple structure, easy to implement and other advantages has become one of the main methods of motor control. However, the pure PID controller often fails when it encounters high-order time-delay systems, highly nonlinear systems and systems that are difficult to establish accurate mathematical models [5]. In addition, the conventional PID controller still has many problems, such as gain sensitivity, sudden load torque resulting in slow running speed, unexpected overshoot and excessive settling time [6]. For the shortcomings of traditional PID controller, PID controller is often combined with other methods to improve the overall performance of the controller.

As an important branch of artificial intelligence technology, reinforcement learning is mainly through the data interaction between agent and environment. Compared with the classical control method, reinforcement learning does not need to obtain accurate dynamic model, which is very advantageous in solving highly nonlinear model. Hu et al. [7] Proposed a diagonal recurrent neural network control method based on Q-Learning in. The weight of diagonal recurrent neural network is updated through online learning of Q-learning, which improves the speed control performance of Brushless DC motor. However, Q-learning can not solve the high-dimensional reinforcement learning task. Depth deterministic strategy gradient algorithm is a deep reinforcement learning algorithm suitable for processing continuous actions [8]. Compared with Q-learning, DDPG has better performance in dealing with continuous reinforcement learning tasks.

This paper aims to establish the control system of Brushless DC motor and improve the speed tracking accuracy. The operation state of BLDCM is monitored by DDPG, so as to compensate the proportional, integral and differential links of PID controller, improve the overall performance of the controller, and use the improved Gaussian function and the maximum critical speed interval to design the reward function, which enhances the versatility.

978-1-6654-3386-0/21/$31.00 ©2021 IEEE

II. MATHEMATICAL MODEL OF BRUSHLESS DC MOTOR

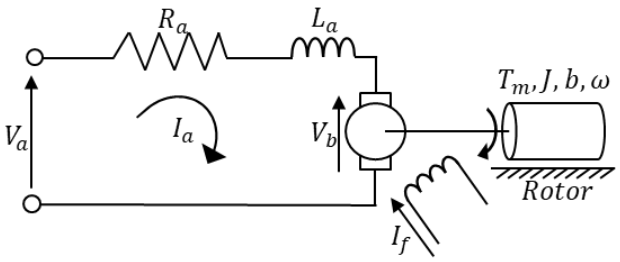


Figure 1. Equivalent circuit diagram of DC motor

The equivalent circuit diagram of Brushless DC motor is shown in Figure 1. In this paper, the motor speed tracking control is carried out by changing the armature voltage. Armature voltage and motor torque can be expressed as follows:

$$V_a(t) = R_a \cdot I_a + L_a \cdot \frac{dI_a(t)}{dt} + V_b(t) \tag{1}$$

$$T_m(t) = J \cdot \frac{d\omega(t)}{dt} + b \cdot \omega(t) \tag{2}$$

Where $V_a(t)$ is the armature voltage, I_a is the armature current, R_a is the armature resistance, L_a is the armature inductance, V_b is the back EMF. T_m is motor torque, J is motor bearing rotation inertia measurement, ω is motor angular velocity, b is fractional constant.

The relationship between motor torque and armature current and back EMF can be expressed as follows:

$$T_m(t) = K_T \cdot I_a \tag{3}$$

$$V_b = K_b \cdot \omega(t) \tag{4}$$

Where K_T is the torque constant, K_b is the back EMF constant.

The motor parameters are shown in Table 1. The transfer function of motor speed and input voltage can be expressed as follows:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_T}{L_a J s^2 + (R_a J + L_a b)s + (R_a b + K_b K_T)} \tag{5}$$

TABLE 1. PARAMETERS OF BRUSHLESS DC MOTOR [9]

Name	Value
Armature inductance(L_a)	$0.1H$
Armature resistance(R_a)	0.45Ω
Rotor inertia(J)	$0.0113N \cdot m \cdot s^2/rad$
Viscous friction coefficient(b)	$0.028\ N \cdot m \cdot s/rad$
Back emf constant(K_b)	$0.067V \cdot s/rad$
Torque constant(K_T)	$0.067N \cdot m/A$

III. PID CONTROL AND DEEP DETERMINISTIC POLICY GRADIENT ALGORITHM

A. PID control

PID controller has the advantages of high efficiency, simple and easy to implement, which has important application in the field of control. In the PID controller, the error signal $e(t)$ is taken as the input and adjusted by the PID controller, the output $u(t)$ is as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{6}$$

WhereK_p, K_i,K_d is the parameter of proportion, integral and differential.

B. Reinforcement learning and deterministic policy gradient method

The basic process of reinforcement learning can be expressed as that the agent in state s_t at any time to perform action a on the environment a_t.The environment will give the agent a reward r_{t+1}, the state changes to the next state s_{t+1}, the future reward value is weighted by the discount coefficient $\gamma(0 \le \gamma \le 1)$, and the cumulative reward R_t after time t is as follows:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{7}$$

Deep reinforcement learning effectively combines deep learning with reinforcement learning. Through limited learning, the agent makes the strategy network and action value network approach to the optimal strategy function and optimal value function, and improves the ability to solve complex reinforcement learning tasks. In DDPG algorithm, the objective function can be defined as the expected value of cumulative reward. In order to maximize the expected value, the strategy μ in the objective function needs to be maximized as follows:

$$J_\beta = \mathbb{E}_\mu[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots] \tag{8}$$

$$\mu^* = \underset{\mu}{\operatorname{argmax}} J(\mu) \tag{9}$$

In most reinforcement learning tasks, the output of strategy function is the probability of action, and the agent needs to select and execute from the probability distribution function of action. When faced with high-dimensional reinforcement learning tasks, this method needs to spend a lot of time. For this problem, the deterministic strategy gradient theorem is proved in [10], as shown in equation (10), which effectively shortens the agent training time.

$$\nabla_{\theta^\mu} J = \mathbb{E}_{s\sim\rho^\beta}\left[\nabla_{\theta^\mu}\mu(s;\theta^\mu)\nabla_a Q(s,a;\theta^Q)|_{a=\mu(s)}\right] \tag{10}$$

Where J is the cumulative discount reward value, and θ^μ and θ^Q are the parameters of strategy function and Q value function respectively.

IV. DDPG-PID CONTROL OF BRUSHLESS DC MOTOR

The control schematic diagram of Brushless DC motor is shown in Figure 2. The main body of the controller is PID, and then it transits to DDPG control.

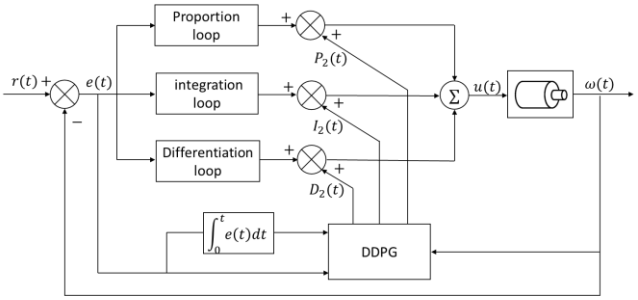


Figure 2. Control diagram of Brushless DC motor

In the control of Brushless DC motor, the error signal $e(t)$ is the difference between the reference signal $r(t)$ and the motor speed $\omega(t)$. DDPG-PID control can be defined as follows:

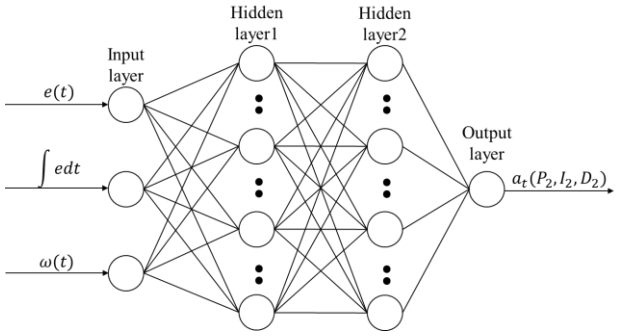
$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \dot{e}(t) + a_t \tag{11}$$

$$e(t) = r(t) - \omega(t) \tag{12}$$

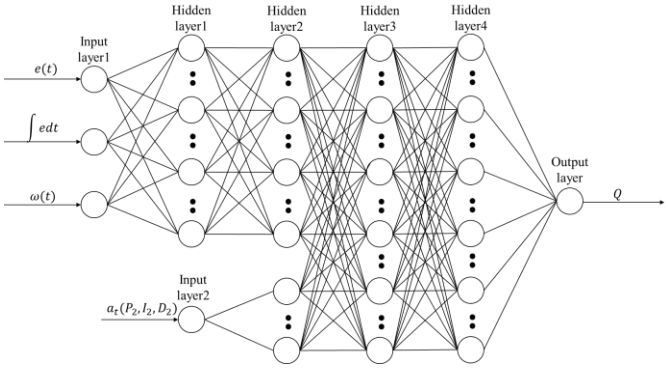
Where $u(t)$ is the output of DDPG-PID controller, a_t is the output of DDPG, which includes the output compensation of proportional, integral and differential parts of PID controller. In the control process, DDPG input speed error, error integral and actual speed as the observation, after a limited number of training to find the optimal strategy, each link of the PID controller output compensation.

A. Network structure of DDPG-PID

In DDPG-PID controller, there are actor network, critic network and corresponding target network. The network structure is shown in Figure 3, and the network parameters are shown in Table 1. In the input layer of the actor network, the input three-dimensional state vectors are the actual speed, speed error and error integral of the brushless DC motor. The middle two hidden layers are full connection layer and active layer, and the output layer is the 3D output of action vector. The input layer of critical network includes three-dimensional state vector and action vector, the middle four hidden layers are full connection layer, activation layer, superposition layer and activation layer respectively, and the output layer is the Q value of action.



(a)



(b)

Figure 3. Network structure (a) actor network (b) critic network

B. Network training of DDPG-PID

In order to make the training data relatively independent, so as to speed up the convergence speed and improve the stability of network update, the data used for network update is not the previous state data obtained by decision-making, but M small batch sample data randomly selected from the experience playback space. The current critic network is updated as follows by minimizing the loss function and using the gradient descent method.

$$Q_{target} = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^{\mu'}\right)|\theta^{Q'}\right) \tag{13}$$

$$L = \frac{1}{M}\sum_{i=1}^{M}\left(Q_{target} - Q(s_i, a_i|\theta^Q)\right)^2 \tag{14}$$

$$\nabla L(\theta^Q) = \frac{1}{M}\left[Q_{target} - Q(s, a|\theta^Q)\nabla_{\theta^Q} Q(s, a|\theta^\mu)\right] \tag{15}$$

Where, Q_{target} is the value of target critic network, $Q(s, a|\theta^\mu)$ is the value of current critic network, and i is the ith sample data.

Current actor network is updated by deterministic policy gradient method as follows:

$$\nabla_{\theta^\mu} J_\beta(\mu) \approx \frac{1}{M}\sum_i \left(\nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s;\theta^\mu)|_{s=s_i}\right) \tag{16}$$

The target critic network and the target actor network are updated by the soft update method with the update rate of τ, as follows:

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \end{aligned} \tag{17}$$

C. Reward function

For most reinforcement learning tasks, the agent will be rewarded or punished after executing the action. When the agent is faced with different states, it will make corresponding actions to get more rewards. Reward function must be universal and provide comprehensive information for agents. Among the problems discussed in this paper, the speed error $e(t)$ is the most concerned variable. If the speed error becomes larger, it needs to be punished, otherwise it will be rewarded. At the same time, in

order to avoid the agent's behavior damaging the motor, the agent's behavior is limited. Therefore, the improved Gaussian function and the maximum critical speed interval are used to define the reward function as follows:

$$r = \alpha r_1 + \beta r_2$$
$$r_1 = e^{\frac{-(e(t)-e_0)^2}{2c^2}} - 0.5 \quad (18)$$
$$r_2 = \begin{cases} 0, & |\omega| < \varepsilon \\ -1, & other \end{cases}$$

Where α and β are reward coefficients, e_0 is the expected error, c is the standard deviation of Gaussian function, and ε is the maximum critical velocity.

D. *The training process of DDPG-PID*

DDPG-PID is an effective combination of DDPG algorithm and PID. The training process is shown in algorithm 1.

Algorithm 1 :DDPG-PID algorithm

Initialize Critic network $Q(a, s|\theta^Q)$ and actor network $\mu(s|\theta^\mu)$

Initialize target network $Q'(\theta^{Q'})$ and $\mu'(\theta^{\mu'})$ with same weights

initialize replay memory RM

initialize a Gaussian Noise G

for episode = $1 \cdots N$ do

Receive initial observation state s_1

for $t = 1 \cdots T$ do

select action $a_t = (P_2, I_2, D_2) = \mu(s_t|\theta^\mu) + G$

select execution action a_t

if $\omega(t) \notin [-\varepsilon, \varepsilon]$

reject a_t and add a negative number to r

else:

execute a_t and get reward r_t and observe new state s_{t+1}

store transition (s_t, a_t, r_t, s_{t+1}) in RM

sample mini-batch of n transitions (s_i, a_i, r_i, s_{t+1}) from RM

set $Q_{target} = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$

update critic according equation (14) and (15)

update actor according equation (16)

update the target networks according equation (17)

end for

end for

First, initialize the online network and target network parameters (Q, μ, Q', μ'), memorize the playback space RM and Gaussian noise G. After receiving the initial observation state s_1, the agent selects action $a_t(P_2, I_2, D_2)$ according to strategy μ and noise G. The reward function will reward each action performed by the agent. At the same time, when the agent executes the action, it will monitor the motor speed in real time. If the actual speed of the motor is within the maximum critical speed range, it will get the corresponding reward after executing the action, so as to continue to observe the next state s_{t+1}.Otherwise, the action is stopped immediately and punished, and the agent reselects the action and executes it. The data (s) formed in this process (s_t, a_t, r_t, s_{t+1}) tuple will be stored in memory playback space RM. Random extraction of small batch tuple data from memory playback space RM.

V. EXPERIMENTS AND RESULTS

In order to highlight the speed tracking effect of DDPG-PID controller on brushless DC motor, the DDPG-PID and PID controller are compared in the simulation environment of MATLAB. The target reference speed is $50\sin(4\pi t)\ r/min$ and the simulation time is $2s$. The PID parameters are the same in the two control methods. In Figure 4, the agent of DDPG-PID controller is in the exploration stage at the initial stage of training, and the reward value is low. However, with the increase of training times, the reward value is more and more, and finally tends to be stable. According to the results of Figure 4, the correctness and stability of DDPG-PID model are verified.

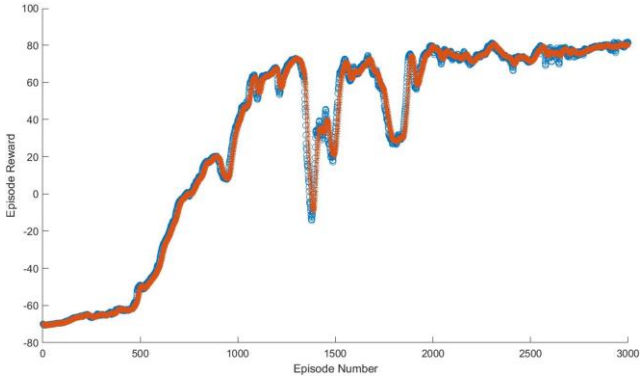


Figure 4. Reward value of agent

Figure 5 and figure 6 respectively show the speed tracking performance of PID and DDPG-PID controller under sinusoidal input. Compared with PID controller, DDPG-PID controller can achieve the desired speed value better.

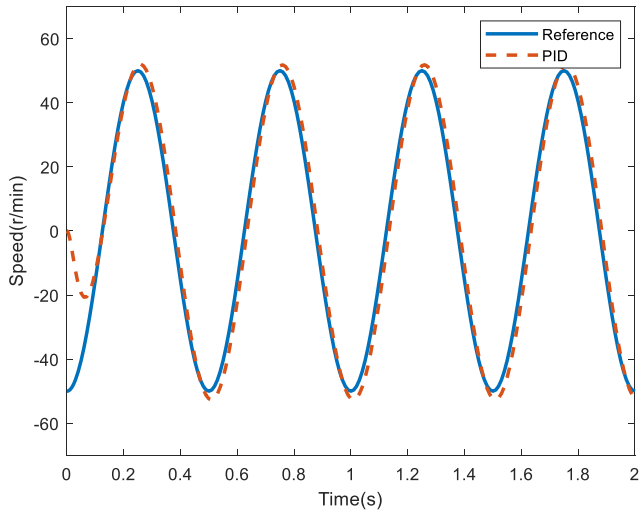


Figure 5. Speed tracking of PID control

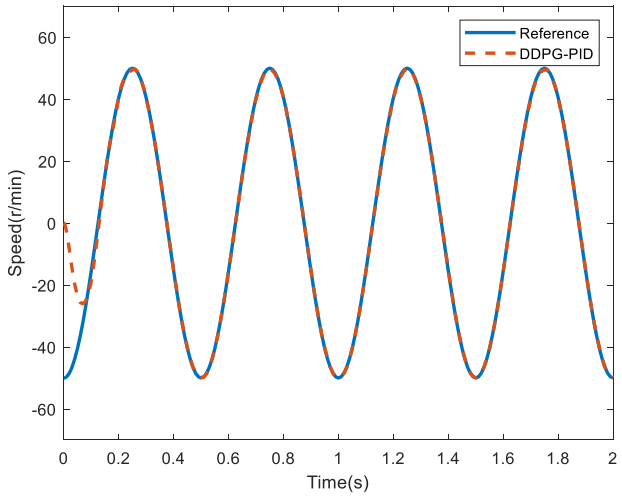


Figure 6. Speed tracking of DDPG-PID control

As can be seen from Figure 7, compared with DDPG-PID, PID controller has larger speed tracking error and oscillation amplitude. On the contrary, the tracking error of DDPG-PID controller is very small and has good stability.

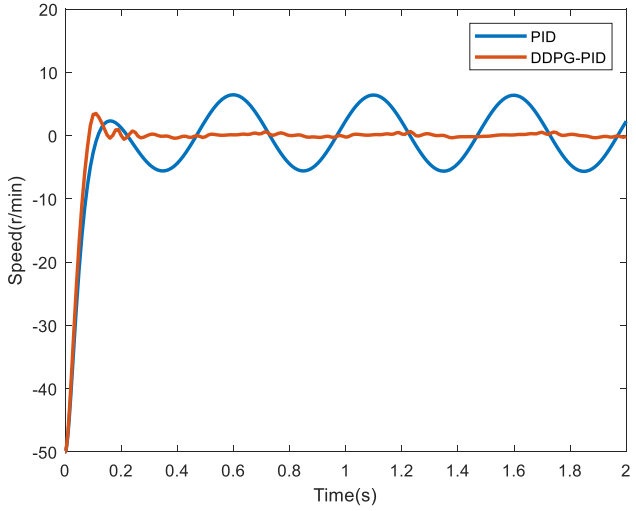


Figure 7. Speed tracking error of PID and DDPG-PID

VI. CONCLUSION

This paper presents a DDPG-PID controller based on DDPG and PID. Based on DDPG and PID, DDPG-PID updates the network by interacting with the environment for a limited number of times to make actor network and critic network approach the optimal strategy function and value function respectively. Secondly, the improved Gaussian function and the maximum critical speed interval are used to reward the behavior of the agent, which enhances the versatility of the algorithm. By using DDPG to compensate the proportional, integral and differential terms of PID controller, the speed tracking error of Brushless DC motor can be effectively reduced. The simulation results show that, compared with PID, DDPG-PID controller has better speed tracking accuracy for BLDCM.

REFERENCES

[1] H. S. Purnama, T. Sutikno, S. Alavandar, A. C. Subrata, and Ieee. (2019) *Intelligent Control Strategies for Tuning PID of Speed Control of DC Motor - A Review. In:* (2019 Ieee Conference on Energy Conversion). pp. 24-30.

[2] H. Suryoatmojo *et al.* (2020) ROBUST SPEED CONTROL OF BRUSHLESS DC MOTOR BASED ON ADAPTIVE NEURO FUZZY INFERENCE SYSTEM FOR ELECTRIC MOTORCYCLE APPLICATION. *International Journal of Innovative Computing Information and Control,* vol. 16, no. 2, pp. 415-428, doi: 10.24507/ijicic.16.02.415.

[3] C. Zheng, Y. Li, and Ieee. (2016) Sensorless Speed Control for Brushless DC Motors System Using Sliding-Mode Controller and Observers. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics*, (International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 216-220.

[4] H. Hu, T. Wang, S. Zhao, and C. Wang. (2019) Speed control of brushless direct current motor using a genetic algorithm-optimized fuzzy proportional integral differential controller. *Advances in Mechanical Engineering,* vol. 11, no. 11, Art no. 1687814019890199, doi: 10.1177/1687814019890199.

[5] J. Karthikeyan and R. D. Sekaran. (2011) Current control of brushless dc motor based on a common dc signal for space operated vehicles. *International Journal of Electrical Power & Energy Systems,* vol. 33, no. 10, pp. 1721-1727, doi: 10.1016/j.ijepes.2011.08.014.

[6] A. Sathyan, N. Milivojevic, Y.-J. Lee, M. Krishnamurthy, and A. Emadi. (2009) An FPGA-Based Novel Digital PWM Control Scheme for BLDC Motor Drives. *Ieee Transactions on Industrial Electronics,* vol. 56, no. 8, pp. 3040-3049, doi: 10.1109/tie.2009.2022067.

[7] H. Hu, T. Wang, H. Wang, and C. Wang. (2020) Q-learning optimized diagonal recurrent neural network control strategy for brushless direct current motors. *Advances in Mechanical Engineering,* vol. 12, no. 9, doi: 10.1177/1687814020958221.

[8] T. P. Lillicrap *et al.* (2015) CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING. In: *Computer ence*.

[9] A. Abdulameer, M. Sulaiman, M. S. M. Aras, and D. Saleem. (2016) Tuning Methods of PID Controller for DC Motor Speed Control. *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 3, no. 2, doi: 10.11591/ijeecs.v3.i2.pp343-349.

[10] D. Silver et al. (2014) Deterministic Policy Gradient Algorithms. *In Proceedings of the 31st International Conference on International Conference on Machine Learning 2014 Beijing, China,* pp. 21–26.