ELEKTROTEHNIŠKI VESTNIK 91(5): 257-264, 2024
ORIGINAL SCIENTIFIC PAPER

Reinforcement learning speed control of a separately excited DC motor

Abdeslam Benmakhlouf[1], **Ghania Zidani**[2], **Djalal Djarah**[1]

[1] *Department of Electrical Engineering, Faculty of Applied Sciences, University of Kasdi MERBAH, Ouargla, 30000, ALGERIA*
[2] *Department of Pharmacy, Faculty of Medicine, University of Batna 2, Batna, 05000, ALGERIA*
E-mail: benmakhlouf.abdeslam@univ-ouargla.dz

Abstract. This paper presents a unique method to control the speed of a separately excited DC motor by leveraging Reinforcement Learning (RL). An RL agent, educated using an Actor-Critic (AC) approach is capable to make effective real-time decisions and evaluations. The agent is trained in the Simulink environment, enriched with Simscape components to enhance the performance. By processing four continuous inputs, i.e. the speed error, its integral, its derivative, and the motor armature current, the agent outputs a PWM signal to control the four H-bridge switches that drive the DC motor. The reward function encourages the agent to learn an optimal control policy that minimizes both the tracking error and the armature current. The simulation results show that the RL-based speed controller outperforms the traditional controllers using cascade PI controllers.

Keywords: DC motor, speed control, reinforcement learning (RL), actor-critic, PI, PSO

Spodbujevalno učenje regulatorja hitrosti ločeno vzbujenega enosmernega motorja

V prispevku predstavljamo metodo za regulacijo hitrosti ločeno vzbujenega enosmernega motorja, ki izkorišča spodbujevalno učenje. Agent spodbujevalnega učenja lahko samostojno sprejema odločitve in ocene v realnem času. Učenje smo izvajali v okolju Simulink s komponentami Simscape za izboljšanje zmogljivosti. Agent kontinuirano spremlja štiri vhode: napako hitrosti, njen integral in odvod ter tok motorja. Na izhodu je pulzno-širinsko moduliran signal za upravljanje štirih stikal H-mosta, ki krmilijo enosmerni motor. Funkcija nagrajevanja spodbuja agenta, da se nauči optimalne strategije vodenja, ki minimizira napako sledenja in tok. Rezultati simulacije potrjujejo, da predlagani regulator prekaša tradicionalne, ki uporabljajo kaskadne regulatorje PI.

1 INTRODUCTION

The Direct Current (DC) motors are prominent electromechanical devices that convert the electrical energy into a mechanical work by exploiting the interactions of the magnetic fields. The interactions are based on the fundamental laws of electromagnetism and the Lorentz force. The DC motors have been widely adopted in various domains due to their low cost, high flexibility, remarkable versatility, and inherent durability [1]. Their speed control is important for ensuring their efficiency and dependability in different operational settings. The primary goal of the DC motor speed controllers is to swiftly reach a desired speed within a predefined reference interval and to robustly cope with external perturbations, such as load variations and parameter changes [2]. However, attaining an optimal performance, especially in the presence of environmental disturbances and system nonlinearities, requires a level of control robustness that surpasses the abilities of the conventional PID controllers, despite their generally satisfactory performance [3]. Therefore, alternative control strategies need to be investigated to tackle the subtle challenges posed by certain operational constraints. Some of the alternative control strategies that have been proposed in the literature include a multiple voltage control [4], Ward Leonard method [5], commutation error compensation strategy [6], and linear quadratic controller [7].

Reinforcement Learning (RL), a distinct subfield of the Artificial Intelligence (AI), enables agents to accomplish specific goals by optimizing a numerical reward signal [8]. This discipline has progressed along three main paths. The first involves the principle of learning via a trial-and-error, a concept rooted in the study of animal learning within the realms of psychology and neuroscience. The second focuses on the problem of an optimal control, conceptualized in the 1950s using a discrete stochastic version of the environment, known as Markovian Decision Processes (MDP) [9]. The model introduces the idea of the dynamical system state and optimal return function, often referred to as the Reward. It also defines the "Bellman equation" to optimize the agent behavior over time, a process known as dynamic programming [10]. The third path pertains to temporal-difference methods which are a prevailing approach in RL [8]. The

Received 15 June 2024
Accepted 24 October 2024

approach has been significantly enhanced by the advent of the Actor-Critic (AC) architecture [11].

Given their capabilities, the RL algorithms are used to handle complex and dynamic systems, such as the DC motors. The intricate task of a precise speed control in the DC motors can be effectively learned by an RL agent. Moreover, the agent can optimize its policy to accommodate external disturbances and system nonlinearities. Numerous studies and works reported in the literature have explored this topic in depth. The studies can be categorized into two main groups: those which optimize the classical control schemes using the RL techniques, and those which employ RL algorithms to supplant them. [12] deploys a fuzzy Q-learning agent to adjust the gains of a PID controller to get a controller that exhibits commendable performance. [13] applies an RL DDPG TD3 actor-critic method to derive adaptive PI constants used to implement a speed control algorithm for a DC motor. Such controller outperforms a PI controller adjusted using conventional methods. [14] uses an RL algorithm to tune a PI controller and compares the obtained results with a PI controller tuned using various optimization methods. [15 employs an Integral RL technique to design a DC motor controller and juxtaposes their design with a traditional RL-based controller. The study underscores the optimality and adaptability of the RL algorithms. [16] introduces a speed controller for a permanently excited DC motor based on a TD3 algorithm. The controller demonstrates an excellent speed tracking performance. [17] presents an application of the Deep Q-Network (DQN) to the speed-tracking control of a DC motor.

Despite the satisfactory performance of the above controllers, the simulation environments used for their development are simplistic control scenarios in accurately representing the industrial and real-world DC motor control schemes. [18] employs an advanced simulation model to train a DDPG agent to learn the optimal control of a DC-DC Buck converter-fed DC motor. The model simulation results underscore the generalization capabilities of the RL agent. [19] presents an online training of an RL agent for DC motor control. The RL agent learns to execute a complex control task based solely on a real-time reward signal, thereby eliminating the need for an offline training.

We use an AC agent to control the DC motor speed. The agent generates a binary discrete action in the form of a PWM signal to control the power switchers of a H-bridge. It is trained in a simulation environment that accurately reflects real-world scenarios of DC motor control. The RL-based controller outperforms the classical controller that employs the PSO fine-tuned twin cascaded PI controllers. The used classical controller exhibits a greater robustness compared to the controller using a single PI controller.

The structure of our paper is as follows. Section 2 provides a concise overview of the RL framework. Section 3 presents a mathematical model of a separately excited DC motor. Section 4 outlines a classical DC motor speed control scheme, employing two PI regulators, with parameters fine-tuned using the PSO algorithm. Section 5, along with its subsequent subsections, delves into the design intricacies of the proposed RL-based DC motor controller. Section 6 presents and analyzes our simulation results. Section 7 draws conclusions of our work and outlines future perspectives for enhancing the presented work.

2 REINFORCEMENT LEARNING

Reinforcement Learning is a Machine Learning approach that is based on a set of fundamental principles, distinguishing itself from other Machine Learning approaches. It is typically performed within a framework of interaction between a learning agent and an initially unknown environment (Fig. 1), modeled as a Markov Decision Process (MDP).

At each time step, the environment produces a state. Upon receiving the current state, the agent responds with an action, computed based on a policy, and subsequently executes it. The action leads to a transition of the environment to a new state. The environment then provides the new state as well as a reward, indicating the quality of the new state. The agent receives the new state representation and the corresponding reward, and the entire process repeats.

The agent consists of two components: a policy and a learning algorithm. The policy is a mapping between the current observation of the environment and a probability distribution of actions to take. Within the agent, the policy is implemented by a function approximator. It has adjustable parameters and uses a specific approximation model, such as a deep neural network (DNN).

The learning algorithm continuously updates the policy parameters based on actions, observations, and rewards. The goal of the learning algorithm is to find an optimal policy that maximizes the long-term expected cumulative discounted reward. RL seamlessly integrates Neural Networks (NNs) and Machine Learning techniques into a problem-solving process, empowering agents to make informed decisions based on unstructured input data. It eliminates the need for a labor-intensive manual engineering of the state space, as agents autonomously determine optimal actions to achieve their objectives.

The RL methodologies frequently leverage NNs to approximate both the policy (actor) and value function (critic). Such use of NNs enables the agent to effectively handle the intricate, high-dimensional input data, which may include images or sensor readings. Consequently, these networks proficiently represent complex policies and value functions, enhancing the agent ability to tackle intricate tasks and decision-making processes.

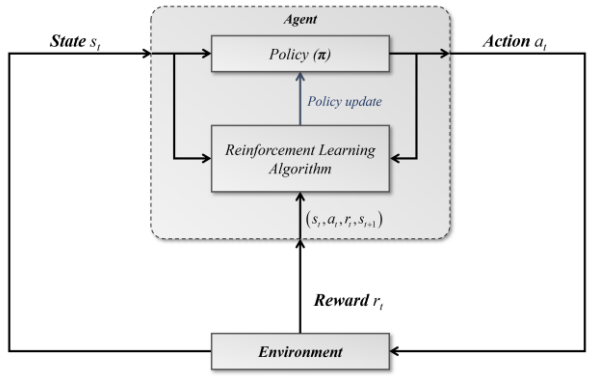

Figure 1. Reinforcement Learning framework.

3 SEPARATELY EXCITED DC MOTOR MODEL

DC motors are widely used as actuators due to their straightforward construction and the comparatively uncomplicated integration of the speed control algorithms. This accounts for their ubiquity in the field of mobile robotics and various industrial sectors, such as manufacturing and transportation. Some of the speed control algorithms that are easy to integrate with the DC motors include Pulse-Width Modulation (PWM), Proportional-Integral-Derivative (PID) control, and Fuzzy Logic Control (FLC).

A separately excited DC motor can be conceptualized as a linear Single Input Single Output (SISO) plant model of the third order. The DC motors have earned a reputation for their exceptional control over the speed and position [20]. There is a simple yet profound mathematical relationship between the angular velocity (ω) of the motor shaft and the input voltage (V_a) applied to the armature. Their relationship can be derived from the fundamental physical laws. This understanding provides a solid basis for a further exploration and application of the DC motor control systems.

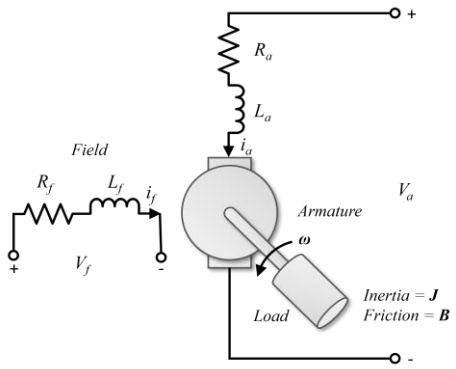

Figure 2. Separately excited DC motor.

The schematic diagram of the DC Motor is shown in Fig. 2.

The following equations describe the dynamic behavior of a DC motor controlled by the armature current. The motor air gap flux (ϕ) is directly proportional to the field current.

$$\phi = K_f i_f(t) \tag{1}$$

The torque produced by the motor is assumed to have a linear relationship with both the air gap flux and the armature current,

$$T_m = K_1 \phi i_a(t) \tag{2}$$

$$T_m = K_1 K_f i_f(t) i_a(t) \tag{3}$$

where K_1 and K_f are constants.

When a steady field current is set up in a field coil, the motor torque is:

$$T_m = K_m i_a(t) \tag{4}$$

The Laplace transformation of Equation 4 is:

$$T_m = K_m I_a(s) \tag{5}$$

The input voltage, when applied to the armature, is linked to the armature current:

$$V_a(s) = R_a I_a(s) + L_a s I_a(s) V_b(s) \tag{6}$$

where $V_b(s)$ is the Back EMF voltage which is proportional to the motor speed. Therefore:

$$V_b(s) = K_b \omega(s) \tag{7}$$

where $\omega(s) = s\theta(s)$ is the transform of the angular velocity and the armature current is:

$$I_a(s) = \frac{V_a(s) + K_b \omega(s)}{R_a + L_a s} \tag{8}$$

The torque produced by the motor is the same as the torque transferred to the load. This can be expressed as:

$$T_m(s) = T_l(s) + T_d(s) \tag{9}$$

In this context, T_l is the load torque and T_d is the disturbance torque, which is typically considered negligible. Therefore:

$$T_l(s) = Js\omega(s) + b\omega(s) \tag{10}$$

Hence, when Td equals zero, the transfer function of the motor is as:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{L_a J s^2 + (L_a b + R_a J)s + (R_a b + K_m K_b)} \tag{11}$$

In this context, the angular velocity ($\omega(s)$) is the output and the armature voltage ($V_a(s)$) is the input.

Table. 1 shows the parameters of the DC motor used in our study.

Table 1. DC motor parameters.

Parameter	Value
R_a: Armature resistance (Ω)	2.581
L_a: Armature inductance (H)	0.025
R_f: Field resistance (Ω)	281.2
L_f: Field inductance (H)	156
L_{af}: Field armature inductance (H)	0.9483
J: Total inertia (Kg.m^2)	0.02215
B_m: Viscous friction coefficient (N.m.s)	0.002953
T_f: Coulomb friction torque (N.m)	0.5161

4 CASCADE PI DC MOTOR SPEED CONTROL

Fig. 3 depicts the architecture of a DC motor speed controller. It consists of two *PI* controllers as inspired by [21]. The PI_1 controller controls the motor speed by taking the speed error as the input and producing the reference armature current. The PI_2 controller controls the current that empowers the motor armature. PI_2 compares the actual current generated by the H-bridge with the reference value and produces a direct voltage signal.

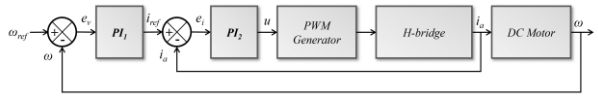

Figure 3. DC motor speed controller.

The PWM block plays a pivotal role in this setup by converting the DC voltage into a binary switching signal. The signal is subsequently used to control the switches within the H-bridge. To be noted, the H-bridge is a vital component in the power electronics, engineered to enable the voltage to be applied across a load in both directions. Its name originates from its distinctive 'H' configuration. The structure consists of four switches. They can be either transistors or Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs). The innovative design of the H-bridge allows the polarity of the voltage applied to the load to be switched, offering a versatile solution for controlling the direction of the current flow. This feature is especially beneficial in applications such as motor driving, where altering the direction can reverse the motor rotation.

The parameters of the two PI controllers are optimized using the Particle Swarm Optimization (PSO) algorithm. The PSO algorithm, proposed by Kennedy and Eberhart [22], draws its inspiration from the principles of the swarm intelligence (SI). Modelled after the collective behavior observed in flocking birds and schooling fish, PSO stands as a potent meta-heuristic global optimization technique. Its robust computational efficiency renders it a versatile tool applicable to a wide array of technical and engineering challenges.

PSO is a population-based, self-adaptive, stochastic optimization technique that operates as follows. The algorithm initiates by generating the initial particles and assigning them initial velocities. It evaluates the objective function at each particle location, determining the best function value and the optimal location. Subsequently, it iteratively updates the particle locations (the new location is the sum of the old location and the velocity, adjusted to keep the particles within bounds), velocities, and neighbors. The iterations continue until the algorithm meets a stopping criterion.

In specific case, the cost function is defined as follows:

$$C = \int_0^t \left(\left| \omega_{ref} - \omega \right| + \left| i_a \right| \right) dt \tag{12}$$

The cost function in Eq. 12 is designed to minimize the tracking error ($|\omega_{ref} - \omega|$) and the control effort. It is expressed by the armature current ($|i_a|$). Executing the algorithm for 10,000 iterations (which is the stopping criterion) yields the following parameters:

$K_{p1} = 1.4261$
$K_{i1} = 195.7448$
$K_{aw1} = 196.4208$
$K_{p2} = 199.8750$
$K_{i2} = 94.5907$
$K_{aw2} = 48.8651$

Where K_p is the proportional gains, K_i is the integral gains, and K_{aw} is the anti-windup gains.

5 RL CONTROLLER DESIGN

The Simulink model, (see Fig. 4), provides a comprehensive representation of the presented control system, incorporating a broad range of constituent elements. Central to this framework is the RL agent which controls the DC motor velocity. The control is accomplished by interpreting the three key inputs: observations, rewards, and a Boolean indicator that signals the end of an episode. Using this information, the RL agent generates a control signal. In this context, it is represented by the PWM signal which guides the operational dynamics of the DC motor.

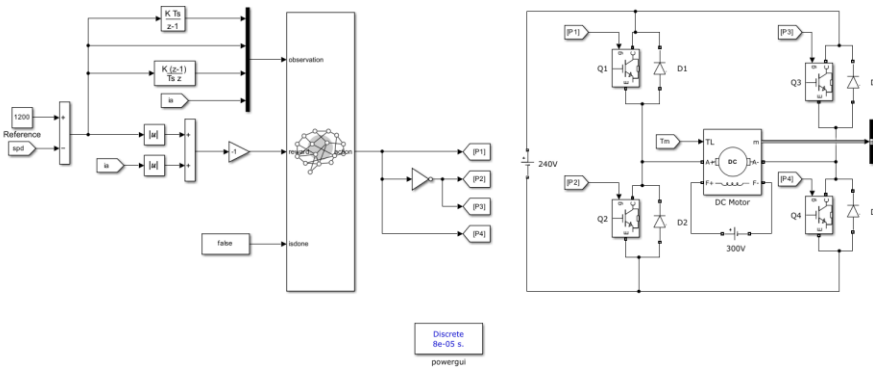

Figure 4. RL speed control system.

The centerpiece of the model is the DC motor block. It is meticulously engineered to encapsulate the electrical and torque dynamics inherent to the separately excited DC motor. The interaction between the RL agent and the DC motor is facilitated by the H-bridge interface, distinguished by its quartet of switching devices, designed to convert a fixed DC input into a variable DC output.

The model also incorporates sensors which are thoughtfully placed to measure critical parameters, such as the motor armature current, torque, and rotational speed. such comprehensive depiction not only highlights the intricate dynamics of the proposed control system, but also delineates the synergistic interaction among the RL agent, DC motor, and ancillary components. This enriched description offers a more detailed understanding of the system's operation and the interplay of its components.

5.1 The AC Agent

The Actor-Critic algorithm is a sophisticated RL methodology that combines the strengths of the policy-based strategies (represented by the Actor) and the value-based strategies (represented by the Critic). This integrated approach is designed to circumvent the inherent limitations observed when these strategies are implemented independently by incorporating the principles of the Policy Gradient and Q-Learning [23].
In this architecture, the Actor determines the course of the action based on the current policy. Simultaneously, the Critic evaluates the quality of the chosen action and provides feedback on the potential adjustments to optimize the future decisions.

This dual role provides a harmonious balance between exploration (probing the unknown areas of the environment) and exploitation (utilizing known information to make decisions), thereby capitalizing on the robustness of both the policy and the value functions. The fundamental principle of the Actor-Critic method is outlined in Algorithm 1.

Algorithm 1: Actor-Critic algorithm [23]

Initialize θ and ϕ;
While *not done* **do**

Sample $\{s_i, a_i\}$ from $\theta_\pi(a|s)$ (run the policy to sample trajectories);

Fit $V_\pi^\phi(s)$ to sampled reward sums;

$$A^\pi(s_i, a_i) = r(s_i, a_i) + \gamma V_\pi^\phi(s_i') - V_\pi^\phi(s_i)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_i^N \nabla_\theta \log \pi_\theta(a_i|s_i) A^\pi(s_i, a_i)$$

Update $\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta)$;

End while

5.2 The Inputs and Outputs of the Agent

The RL agent has four inputs: the error, its integral, its derivative, and the armature current. These observations play a pivotal role in specifying and defining the state of the system. The RL agent produces a binary signal as the output, which represents the PWM signal used to control the four switching devices of the H-bridge. The H-bridge, in turn, modulates the DC voltage applied to the armature of the DC motor (Fig. 4). The reference speed follows a constant pattern that varies randomly in the amplitude for each episode. The initial speed of the motor is set to 0 rpm at the start of each episode.

5.3 The Agent Neural Networks

The Neural Networks can be utilized for the nonlinear approximation of the both value and the policy functions in RL. Our network architecture of the AC agent is shown in Fig. 5. It consists of two distinct networks: a value function network (known as the Critic network) and a policy function network (known as the Actor network).

Each layer within the networks is characterized by its connection type (fully connected: FC), the number of

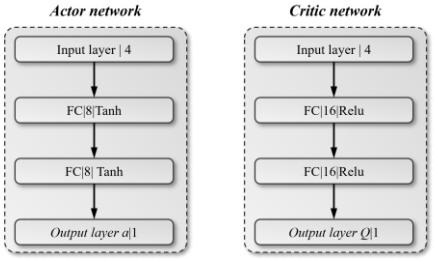

Figure 5. The agent NNs.

the neurons, and the activation function, with the exception of the input layer. In the Actor network, the hyperbolic tangent (Tanh) activation function is used, while the Rectified Linear Unit (ReLU) function is employed in the Critic network. The configuration ensures the optimal performance of the AC algorithm in our scenario.

5.4 The Reward Function

A reward function is a mapping from the state-action pairs to real numbers, signifying the immediate reward the agent receives upon executing a specific action in a given state. The instantaneous reward is calculated using the following equation:

$$r(t) = -\left(|e(t)| + |i_a(t)|\right) \tag{13}$$

where *e(t)* is the instantaneous error, and $i_a(t)$ is the armature current. The cumulative reward is then determined by summing up the instantaneous rewards across the entire episode duration. Maximizing the cumulative reward leads to the simultaneous minimization of both the tracking error and the control effort.

5.5 The Training

Fig. 6 presents a comprehensive depiction of the learning trajectory of the AC agent which has been meticulously trained within the simulated environment delineated in Fig. 4. The light blue curve is the reward acquired from individual episodes and the dark blue curve is the rolling average reward computed over a window of 50 episodes.

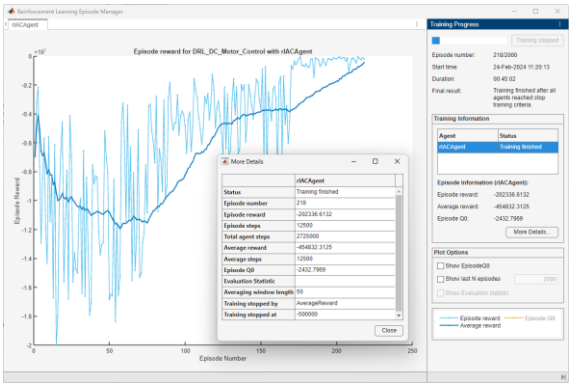

Figure 6. Learning curves.

A significant observation drawn from this data pertains to the convergence of the AC agent towards the target average reward. The remarkable achievement is realized following a sequence of 218 episodes, necessitating a cumulative duration of 45 minutes and 2 seconds. This underscores the efficacy of the AC agent in optimizing its performance within the provided simulated environment.

During training, the reference speed is varied randomly in the range [−1500,1500] rpm at no disturbance applied.

The parameters of the training process are shown in Table. 2.

Table 2. Training process parameters.

Parameter	Value
Sampling time	0.00008 s
Episode time	1 s
Learning rate	0.0001
Discount factor	0.997
Stopping average reward	-500000
Averaging window length	50
Maximum number of the episodes	2000

6 SIMULATION RESULTS

To assess the performance of the designed RL controller, an initial simulation test of the control system is conducted. The simulation takes one second at a constant reference speed of 1200 rpm. At the 0.25 s simulation, an additional input disturbance in the form of a step waveform of a magnitude of -100 N.m is introduced. The effectiveness of the RL control strategy is then evaluated by comparing its performance with that of the cascaded PSO fine-tuned PI-based controller. Fig. 7 shows the step response of the RL controller juxtaposed with that of the PI controller. The comparison sheds light on the proficiency of both controllers in tracking the reference setpoint and mitigating the disturbances. Moreover, Fig. 8 provides an insight into the variation of the armature currents during the control simulation.

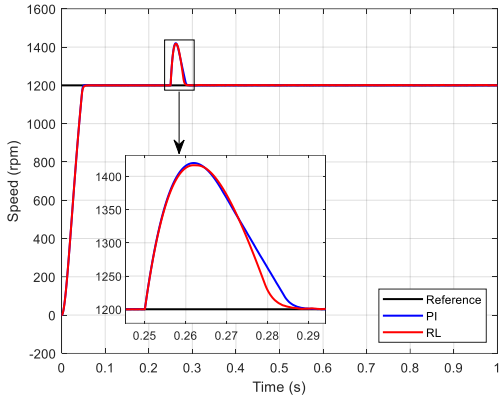

Figure 7. Simulation test 1.

One of the important advantages of the RL controller is its adaptability in navigating diverse simulation environments which include randomized setpoints. Notably, despite the absence of explicitly applied disturbances during the RL agent training phase, its experience-based learning enables it to develop robust responses to input disturbances.

In this context, it should be noted that the PI controller demonstrates comparable performance to the RL controller.

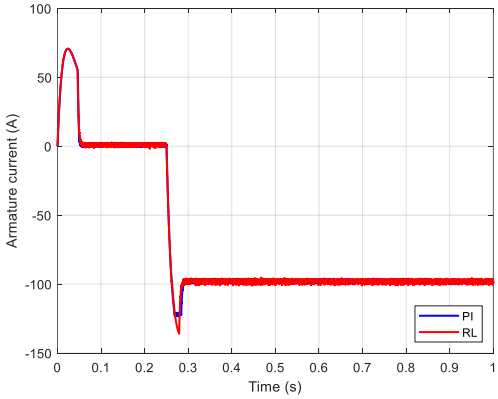

Figure 8. Armature current variations.

Figs. 9 and 10 depict two additional simulation tests aimed at tracking different input signals. In Fig. 9, there is no overshoot observed with the RL controller. Conversely, Fig. 10 highlights the advantageous capability of the RL controller in effectively tracking a rapidly changing input signal.

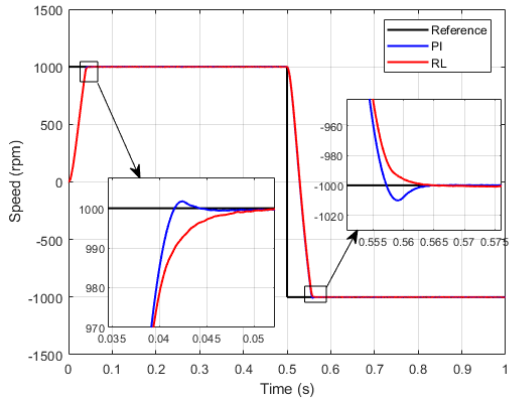

Figure 9. Simulation test 2.

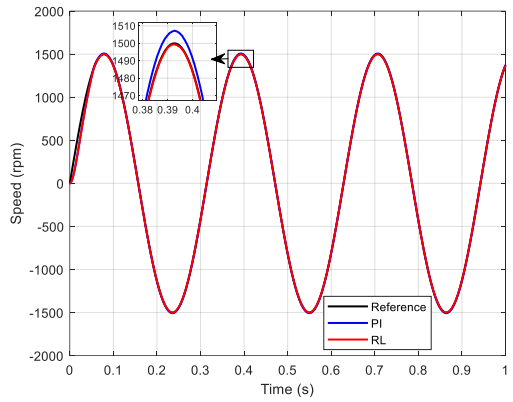

Figure 10. Simulation test 3.

7 CONCLUSION

In our study, an RL algorithm is employed to design a speed controller for a separately excited DC motor. To ensure a superior control performance, a simulation environment is meticulously designed in Simulink. This environment allows for a precise system modeling, incorporating a complete DC motor control scheme and the H-bridge. An AC agent is trained focusing on two control performance objectives as defined by the reward function. The AC agent learns to map the input items to the output binary PWM signal, driving the DC motor towards the desired reference speed. The learning agent replaces the control and the PWM generator of the classical control scheme. The training process is efficient, with a satisfactory performance achieved in just 45 minutes, without the need for numerous episodes.

Our simulation results show that the performance of the RL controller is either comparable or superior to the classical speed controller which uses cascade PI controllers whose parameters are optimized using the PSO algorithm.

In our future work, we shall investigate the real-time implementation of the presented control scheme.

REFERENCES

[1] Abut, Tayfun (2016) Modeling and Optimal Control of a DC Motor, International Journal of Engineering Trends and Technology, 32 (3), 146-150, https://doi.org/10.14445/22315381/IJETT-V32P227

[2] Sachit, S., Vinod, B.R (2022) MRAS Based Speed Control of DC Motor with Conventional PI Control — A Comparative Study, International Journal of Control, Automation and Systems, 20 (1), 1-12, https://doi.org/10.1007/s12555-020-0470-1

[3] Freitas, John Breno Santos Marquezan, Lucas de Oliveira Evald, Paulo Jefferson Dias Peñaloza, Elmer Alexis Gamboa Cely, Marlon Mauricio Hernandez (2024) A fuzzy-based Predictive PID for DC motor speed control, International Journal of Dynamics and Control, 12 (1), 1-11, https://doi.org/10.1007/s40435-023-01368-2

[4] Verdelho, P. and Marques, G.D (1998) DC voltage control and stability analysis of PWM-voltage-type reversible rectifiers, IEEE Transactions on Industrial Electronics, 45 (2), 263-273, https://doi.org/10.1109/41.681225

[5] Choudhary, Abhishek and Singh, Shiv Aishwarya and Malik, Mohd. Faizan and Kumar, Anil and Pathak, Mukesh Kumar and Kumar, Vinod. (2012, January 3-5). Virtual lab: Remote access and speed control of DC motor using Ward-Leonard system. IEEE International Conference on Technology Enhanced Education (ICTEE). Amritapuri, India. https://ieeexplore.ieee.org/document/6208666

[6] Wang, Lu and Zhu, Z. Q. and Bin, Hong and Gong, Liming (2021) A Commutation Error Compensation Strategy for High-Speed Brushless DC Drive Based on Adaline Filter, IEEE Transactions on Industrial Electronics, 68 (5), 3728-3738, https://doi.org/10.1109/TIE.2020.2984445

[7] Yuan Zhi and Wang Weiqing and Cheng Jing and Navid Razmjooy (2022) Interval linear quadratic regulator and its application for speed control of DC motor in the presence of uncertainties, ISA Transactions, 125, 252-259, https://doi.org/10.1016/j.isatra.2021.07.004

[8] Richard S. Sutton Andrew G. Barto (2018). Reinforcement Learning: An Introduction. 2nd Ed. MIT Press.

[9] Richard Bellman (1957) A Markovian Decision Process, Journal of Mathematics and Mechanics, 6 (5), 679-684, https://doi.org/10.1109/TIE.2020.2984445

[10] Puterman, Martin L (1994). Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley. https://doi.org/DOI:10.1002/9780470316887

[11] Konda, Vijay and Tsitsiklis, John (1999). Actor-Critic Algorithms. In S. Solla and T. Leen and K. Müller (Eds), Advances in Neural Information Processing Systems 12 (pp. 1008-1014). MIT Press.

[12] Kofinas, Panagiotis and Dounis, Anastasios I (2018) Fuzzy Q-Learning Agent for Online Tuning of PID Controller for DC Motor Speed Control, Algorithms 11 (10), 1-13, https://doi.org/10.3390/a11100148

[13] Alejandro-Sanjines, Ulbio and Maisincho-Jivaja, Anthony and Asanza, Victor and Lorente-Leyva, Leandro L. and Peluffo-Ordóñez, Diego H (2023) Adaptive PI Controller Based on a Reinforcement Learning Algorithm for Speed Control of a DC Motor, Biomimetics, 8 (5): 1-26, https://doi.org/10.3390/biomimetics8050434

[14] Sevilay Tufenkci, Baris Baykant Alagoz, Gurkan Kavuran, Celaleddin Yeroglu, Norbert Herencsar, Shibendu Mahata (2023) A theoretical demonstration for reinforcement learning of PI control dynamics for optimal speed control of DC motors by using Twin Delay Deep Deterministic Policy Gradient Algorithm, Expert Systems with Applications, 213 (Part C): 1-16, https://doi.org/10.1016/j.eswa.2022.119192

[15] G. Bujgoi and D. Sendrescu. (2022, May 29 – June 1). DC Motor Control based on Integral Reinforcement Learning. International Carpathian Control Conference (ICCC), Sinaia, Romania. https://ieeexplore.ieee.org/document/9805935

[16] B. Cosmin and T. Sorin. (2023, June 29-30). Reinforcement Learning for a Continuous DC Motor Controller. International Conference on Electronics, Computers and Artificial Intelligence (ECAI). Bucharest, Romania. https://ieeexplore.ieee.org/document/10193912

[17] Rossi, Federico and Gruosso, Giambattista and Gajani, Giancarlo Storti. (2023, July 6-8). A Reinforcement Learning based controller for optimal speed control of a DC motor using deep Q-network algorithm. International Conference on Smart Technologies (EUROCON). Torino, Italy. https://ieeexplore.ieee.org/document/10198954

[18] R. Anugula and S. P. Krishna Karri. (2021, Dec 17-19). Deep Reinforcement Learning Based Adaptive Controller of DC Electric Drive for Reduced Torque and Current Ripples. International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES). Raipur, India. https://ieeexplore.ieee.org/document/9751630

[19] Bibek Poudel and Thomas Watson and Weizi Li. (2022, Sep 18 – Oct 12). Learning to Control DC Motor for Micromobility in Real Time with Reinforcement Learning. International Conference on Intelligent Transportation Systems (ITSC). Macau, China. https://arxiv.org/pdf/2108.00138

[20] Dhanesh K. Sambariya and Deepak Paliwal (2016) Comparative Design and Analysis of PIDA Controller Using Kitti's and Jung-Dorf Approach for Third Order Practical Systems, British Journal of Mathematics & Computer Science, 16 (5), 1-16, https://doi.org/10.9734/BJMCS/2016/26223

[21] A. S. Yustin, H. Nugroho and W. K. Wibowo. (2021, June 26-26). Precise Speed Control of DC Motor by Implementing Cascade PI Controller. IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS). Shah Alam, Malaysia. https://ieeexplore.ieee.org/document/9495889

[22] J. Kennedy and R. Eberhart. (1995, Nov 27- Dec 1). Particle swarm optimization. International Conference on Neural Networks. Perth, WA, Australia. https://ieeexplore.ieee.org/document/488968

[23] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu. (2016, June 19-24). Asynchronous methods for deep reinforcement learning. International Conference on International Conference on Machine Learning. New York, NY, USA. https://arxiv.org/abs/1602.01783

Abdeslam Benmakhlouf obtained an engineering degree in automatic control, and a magister degree in robotics from the university of Batna 2 (Algeria). He obtained a PhD in electrical engineering from the university of Ouargla (Algeria), where he is working as a lecturer at the department of electrical engineering. His research interest includes mobile robotics, intelligent control systems, and reinforcement learning.

Zidani Ghania obtained her the diploma of state engineer in electronics, a magister degree and a PhD in robotics from the university of Batna 2 (Algeria). She is currently working as a lecturer at the department of pharmacy (university of Batna 2). Her research interest includes mobile robotics and intelligent control systems.

Djarah Djalal holds a PhD in Electrical Engineering. He is working as a lecturer at the Department of Electrical Engineering at the University of Ouargla, Algeria. His research interests include mobile robotics, artificial intelligence, and deep learning.