Data-Driven Control of Electrical Drives: A Deep Reinforcement Learning with Feature Embedding

Xing LIU, Dengyin JIANG, and Chenghao LIU

***Abstract*—Classical model-based control solutions dominated the research field of numerous electrical drives applications in the past forming a strong basis, since they usually result in control approaches with excellent performance. However, the design of these controllers strongly depends on the available knowledge of the controlled plant, which often leads to the lack of robustness owing to model-dependent nature. To take account of the defect, this work aims to provide a control framework that combines intelligent data-driven-based control protocol with the deep reinforcement learning technique for electrical drives. Specifically, the two key features of this developed control framework that, first, a data-driven control architecture along with deep reinforcement learning technique that embedding the features of the agents' inputs is developed to enhance the performance, second, the environment for the current agent is reformulated so as to avoid mutual interference between the two agents (controllers) in training for both speed and current in a dual-loop system. Finally, we demonstrate our solution and highlight its superiority on a case study, and the results presented are promising and motivate further research in this field.**

***Index Terms*—Feature embedding, intelligent control, motor drives, permanent magnet synchronous motor (PMSM), reinforcement learning (RL).**

I. Introduction

In recent decades, permanent magnet synchronous motor (PMSM) drives, due to their prominent merits, such as high energy efficiency, high power density, high reliability, and wide speed control range, have gained tremendous spread in various industrial and automotive applications. In particular, its applications are expanding rapidly, such as industrial robots, electric vehicles, industrial automation, and power transmission systems. Various control techniques for PMSM drives have been extensively explored in the literature [1]–[3]. Among them, the well-developed method used to control PMSM drive system is proportional-integral (PI) regulator together with pulse width modulation (PWM) modulator designed in continuous-time. To be specific, the PI controller regulates the system state variable to track its desired value by generating a reference voltage signal to the PWM stage. This approach has a simple structure and easy to implement which uses a feedback loop to adjust the control signal based on the difference between the desired and actual values of the motor speed and currents. Although this is a reasonable approach, the controller parameters have to be properly tuned to ensure fast transient response and less steady-state errors.

A. Literature Review and Motivation

Recently, model predictive control (MPC) is receiving considerable attention in electric drive systems [4], [5]. The popularity of this solution stems largely from the possibility to explicitly address multivariable nonlinear systems constraints. In electrical drives, MPC solutions can be loosely categorized in two categories based on whether a modulation stage is needed or not. Continuous control-set MPC (CCS-MPC) produces the continuous-time control inputs to the modulation stage in the controller formulation [6], while finite control-set MPC (FCS-MPC) replaces the cascaded control structure without the intervention of the intermediate stage [7]. The CCS-MPC method can be favored in applications where keeping a fixed switching frequency is crucial. The latter offers an improved dynamic performance which rely on a sophisticated mathematic model to predict future behavior of system state and optimize the control signal. Although extensive research works in the control of CCS-MPC and FCS-MPC have brought some improvements, pursuing excellent control performance while ensuring safety and reliability in the presence of uncertainties remains open for PMSM drive system [8], [9].

The rapid development of data-driven algorithm has provided new avenues to overcome the aforementioned inherent limitation for control system design [10]–[12]. Its main workflow is to obtain the parameters of a function approximator (is

Manuscript received April 2, 2025; revised May 14, 2025 and September 16, 2025; accepted October 12, 2025. Date of publication December 30, 2025; date of current version December 2, 2025. This work was supported by State Key Laboratory of Highspeed Maglev Transportation Technology under the grant SKLM-SFCF-2023-020, National Natural Science Foundation of China under the grant 52577218, Key Laboratory of Special Machine and High Voltage Apparatus (Shenyang University of Technology), Ministry of Education under the grant KFKT202403, Open Research Project of the State Key Laboratory of Industrial Control Technology under the grant ICT2025B53, Open Research Project of the State Key Laboratory of Power Transmission Equipment Technology under the grant SKLPET-kfkt202406, and the key Laboratory of Marine Power Engineering and Technology of Ministry of Transport under the grant KLMPET2025-04. (*Corresponding authors: Dengyin Jiang and Chenghao Liu.*)

X. Liu is with the College of Electrical Engineering, Shanghai Dianji University, Shanghai 201306, China, and State Key Laboratory of Power System, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China, and College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: xingldl@zju.edu.cn).

D. Jiang is with the College of Electrical Engineering, Shanghai Dianji University, Shanghai 201306, China (e-mail: jiangdy@sdju.edu.cn).

C. Liu is with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: liuchenghao@zju.edu.cn).

Digital Object Identifier 10.24295/CPSSTPEA.2025.00037

usually a neutral network) by training it with a large amount of data based on observable variables of the controlled plant, and apply it in control process. The data-based supervised method can be utilized to tune controller parameters, calculate nonlinear magnetic flux, or identify motor parameters. It is even possible to train a neutral network to imitate the output of a PI or an MPC controller [13]. However, during the supervised training, both input and output data for an approximator are required in these applications, thus it is not possible to directly obtain a controller through this method.

Subsequently, with respect to another line of research, reinforcement learning (RL) has been predominantly studied in the electric drive field for many years and it has attracted much attention from researchers [14], [15]. Its workflow is to use the output of a random actor to interact with the environment, and train the actor through the rewards (the output of a critic) generated by the interaction. In this approach, the critic and actor are both neutral networks. The critic network approximates the reward through supervised learning, while the actor network maximizes the reward through exploration and feedback. For general problems in finite action space, deep-Q-network (DQN) can achieve excellent performance, which makes RL applicable for controlling inverter switch states [16]. However, for a complete motor control system, continuous control may be more suitable, especially for the speed-to-current conversion, which involves continuous input and output. It is worth remarking that deep deterministic policy gradient (DDPG) has solved the problems of implementing RL in continuous action space. It is an actor-critic, model-free algorithm based on the deterministic gradient which employs some methods such as experience reply, target network, and soft updates to improve training stability and solve physics tasks robustly. The emergence of DDPG enables the possibility of directly learning the controllers for electric drive systems solely through data-driven approaches.

Some research studies on using RL controllers in PMSM drive system have been devoted to the enhancement of the robustness against parameter mismatch and disturbances. An RL current controller is proposed in [17], where the concept of PMSM controller design by DDPG is first proved. The authors in [18] exploited an RL torque controller by deploying a complex reward rule so as to make the operating point adhere to maximum torque per current strategy. The results in [19] leverage an RL speed controller to reject active disturbance. Overall, for PMSM speed (or torque) control, it is possible to use a single RL controller to track the reference. However, in this sense, the control strategies for torque and current, such as maximum torque per ampere (MTPA) or maximum torque per voltage (MTPV), cannot be guaranteed, unless a complex multi-objective reward can be designed. On the other hand, it is still uncertain how to train two independent RL controllers for the dual-loop system and ensure their convergence while maintaining a strategy module for torque and currents. Motivated by these issues, it is expected to exploit a data-driven control architecture along with deep RL technique that embedding the features of the agents' inputs for electrical drives in a dual-loop system. This consideration encourages the main innovation of the current research.

B. Main Contribution

Pursuing the aforementioned observations, we will launch a crucial study on the deep RL control problem, and we hope that this work lays a theoretical foundation and also inspires new achievements in the intersection of artificial intelligence (AI) technique and deep learning control theory. In this work, we further focus on investigating a novel intelligent data-driven control architecture together with two RL controllers that embedding the features of the agents' inputs for a dual-loop control system. This implies that both the speed controller and current controller are entirely learned by intelligent agents, rather than being designed through model-based approaches. To avoid mutual interference between the two controllers, this paper adopts a sequential training method, where the current controller is trained first, followed by the training of the speed controller. In particular, the current agent interacts only with the inner loop during training. The convergence and performance of the controllers have been validated under different operating conditions. The performance evaluation shows that the RL dual-loop controllers can achieve desired performance to the model-based approaches [17], [18], while also demonstrating better dynamics. Furthermore, this paper leverages embedding techniques to the controller variables, significantly enhancing the accuracy of reference tracking compared to [20], and this method can be naturally extended toward various control systems. Finally, extensive investigations for the electrical drives confirm the interest and the viability of the proposed design methodology.

Compared to existing literature, our method has the following novel aspects.

- Building upon the RL control protocol, in contrast to previously known results, this work goes one step further and accomplishes both speed and current control in a dual-loop PMSM drive system relying solely on data. To be more precise, the sequential training method used in this article avoids mutual interference between the two controllers and effectively aids in the convergence.
- Unlike much prior studies, by transforming the variables into embeddings before inputting them into the controllers, significant improvements are achieved in both training and testing, which facilitate the alleviation of performance deterioration. This modification is quite general and easy to implement in engineering applications and can be conveniently extended to other RL controllers, without sacrificing the simplicity of the control structure.
- Last but not least, effectiveness and performance of the proposal are validated extensively and highlighted by benchmarking it against other state-of-the-art control approaches including RL-based controller and PI-based controller. The extensive testing results in this paper indicate that when designing intelligent controllers, using data from various operating conditions in training is crucial. This literature opens up even more possibilities of

connections with power converter and/or motor control fields.

C. Outline of the Article

The remaining parts are structured as follows. In section II, we briefly describe the PMSM dual-loop drive system. Section III presents the speed and current controllers designed formulation. Meanwhile, we provide a distinctive alternative and details of the proposed RL methodology. To be specific, we exploit an intelligent data-driven-based controller along with deep RL technique. In the following, to further enhance performance under different operation scenarios, our work further focuses on developing a method that embedding the features of the agents' inputs. Further, in order to avoid mutual interference between the two agents (controllers) in training, the environment for the current agent is reformulated for both speed and current in a dual-loop system. In section IV, we verify its merits with different benchmark examples from the literature. Finally, conclusions and future works on the suggested control protocol are summarized in Section V.

II. Physical System

The iteration of an RL intelligent agent is realized through interaction with its environment. The iteration of an RL intelligent agent is realized through interaction with its environment. To address a specific problem, it is crucial to determine the environment in which the agent operates. In this paper, the environment in which the two agents (current agent and speed agent) operate includes physical systems such as a PMSM, an inverter module and a module for selecting operating point. Note that the two agents belong to each other's external environment. This aspect will be discussed in the next session.

In a vector control system, a PMSM can be modeled by a set of differential equations, which is described in the d/q coordinate. It yields:

$$u_d = Ri_d + L_d\frac{\mathrm{d}i_d}{\mathrm{d}t} - \omega L_q i_q \tag{1}$$

$$u_q = Ri_q + L_q\frac{\mathrm{d}i_q}{\mathrm{d}t} + \omega(L_d i_d + \psi_\mathrm{f}) \tag{2}$$

$$T_\mathrm{e} = \frac{3}{2}p\left(\psi_\mathrm{p} i_q + (L_d - L_q)\, i_d i_q\right) \tag{3}$$

$$J\frac{\mathrm{d}\omega}{\mathrm{d}t} = T_\mathrm{e} - T_\mathrm{m} \tag{4}$$

where u_d, u_q, i_d, and i_q are the voltage and current of the motor, T_e and T_m represent the electromagnetic torque and load respectively, and ω represents the machine velocity. All variables in the equations are derived from observation and measurement of the PMSM, and the d/q components are obtained through coordinate transformation. In the completed trained control system, torques (T_e, T_m) are not required to be measured due to the fact that their effects are reflected in changes in velocity. However, during the training phase, they needed to be observed so as to calculate the rotational speed in the environment.

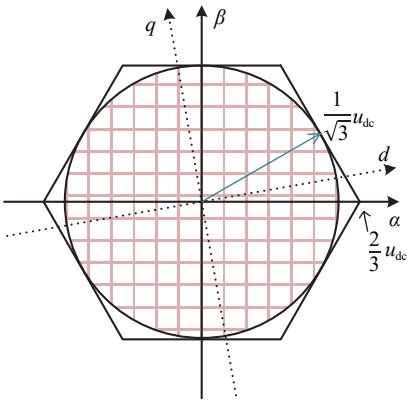

Fig. 1. Limitation of voltage vector.

A power electronic converter is usually deployed to drive the three-phase PMSM. The converter is powered by a constant DC bus and, hence, the voltage range at the stator of the PMSM is limited. When it observed in α/β coordinate in Fig. 1, the maximum voltage that the inverter can provide is limited in a hexagon. Accordingly, even though the output of the RL current controller is (–1,1), the modulation module still limits the actual voltage within a feasible range. This is a characteristic of the environment, which the RL agent needs to implicitly learn. Moreover, it is interesting to remark that the dead time of converter should be taken into account to ensure that RL actor can adapt to the real environment [14].

In order to make the PMSM operate at the optimal point, the desired control solution are necessary after decoupling control into d/q coordinates. The selection of the operating point must firstly satisfy the constraints on current and voltage. High current can lead to temperature rise and safety issues and, hence, the current should be limited:

$$\sqrt{i_d^2 + i_q^2} \leqslant i_\mathrm{max} \tag{5}$$

As for voltage constraint, the operating point voltage must satisfy the requirement that it can be achieved by the modulation module throughout the complete period. It should be noted that, unlike the limit in the inverter, the constraint here refers to the inscribed circle of the hexagon in Fig. 1, with a maximum value of $\frac{1}{\sqrt{3}}u_\mathrm{dc}$. According to (1) and (2), the voltage of PMSM is small at low speed, so the voltage limit can be ignored. At high speed, after neglecting the voltage drop across the resistance, this constraint can be expressed as:

$$\sqrt{(\omega(L_d i_d + \psi_\mathrm{f}))^2 + \omega(L_q i_q)^2} \leqslant \frac{1}{\sqrt{3}}u_\mathrm{dc} \tag{6}$$

After satisfying the current and voltage limits, the suggested strategy for the operating point is generally based on MTPA and MTPV. Based on (3) and the Lagrange Multiplier method, the reference values of d/q axis currents under MTPA strategy can be obtained:

$$i_d = -\frac{\psi_\mathrm{f}}{2(L_d - L_q)} + \sqrt{\frac{\psi_\mathrm{f}^2}{4(L_d - L_q)^2} + i_\mathrm{q}^2} \tag{7}$$

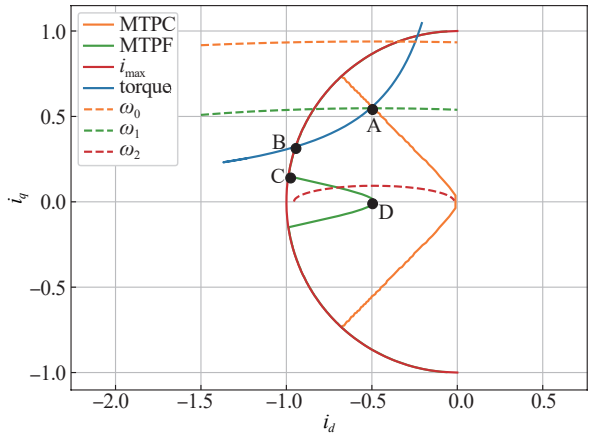

Fig. 2. Operating point selection strategy.

Similarly, the reference values under MTPV strategy are given as:

$$\begin{cases} i_d = \dfrac{3\psi_f L_q - 4\psi_f L_d + \sqrt{\psi_f^2 + \dfrac{8}{3}u_{dc}^2\left(\dfrac{L_d - L_q}{\omega L_q}\right)^2}}{4L_d(L_d - L_q)} \\ i_q = -\dfrac{i_q}{\omega L_q} \end{cases} \tag{8}$$

The operating point selection strategy is illustrated in Fig. 2, which aims to achieve the maximum torque while satisfying (5) and (6). For the reference torque shown in the figure (blue line), when the speed is low, the motor operates at point A determined by MTPA. Since the speed increases beyond the critical point of the voltage limit, the system will subsequently operate along the constant torque curve until point B. At this point, the current is saturated, and the PMSM is unable to maintain a constant torque. The motor will work along the current limit circle to provide maximum possible torque. Finally, the critical point of MTPV is reached. If the motor continues to accelerate, it will subsequently operate along the point C to D curve to fully utilize the voltage and obtain the maximum torque. The point D is a theoretical point at $(-\frac{\psi_f}{L_d}, 0)$. During training, the operating point is obtained through look-up-table (LUT) or analytical method. This strategy module, like the PMSM physical system, also belongs to the environment that the RL agent needs to adapt.

III. Design and Training

In this section, motivated by the aforementioned discussions, we aim at investigating on an intelligent data-driven-based controller design issues. To this aim, a data-driven control architecture along with deep RL technique that embedding the features of the agents' inputs is presented to enhance the performance. Meanwhile, the environment for the current agent is reformulated so as to avoid mutual interference between the two agents (controllers) in training for both speed and current in a dual-loop system for electrical drives. In what follows, the suggested control design procedure will be discussed in detail.

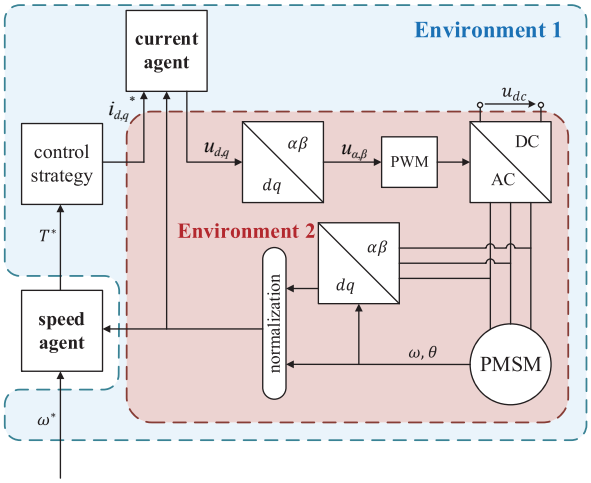

Fig. 3. The RL agents and their environments.

A. Design Training Algorithm

The overall system is shown in Fig. 3, where the blue box represents the environment of speed agent, and the burgundy box represents the environment of current agent. Ideally, the Environment 2 should include all components in the system except the current agent. However, in this sense, updating the parameters of the speed agent will lead to change in the environment of the current agent, which will require retraining. Similarly, variations in the parameters of the current agent will require retraining of the speed agent.

On the other hand, if training both agents synchronously, it may lead to instability due to their different objectives and the mutual influence of their convergence. To address the issues associated with the aforementioned methods, this paper leverages a solution where the current controller is first trained in Environment 2, followed by training the speed controller in Environment 1, and as shown in Fig. 3. Since the current loop is an inner loop, its dynamics should be faster than speed loop. Consequently, when the current controller is trained, the environment of the current agent can be simplified by neglecting the dynamic process of the outer loop, as long as the trained current controller can track the references under different speed conditions.

The RL controller is actually the actor of the agent, whose objective is to maximize the reward. For the both speed and current RL controllers, their rewards are the negative mean squared error (MSE) between the actual value and the reference value. Then, we can get

$$r_{\text{speed}} = -\left(\frac{\omega^* - \omega}{\omega_n}\right)^2 \tag{9}$$

$$r_{\text{current}} = -\left(\frac{i_d^* - i_d}{i_n}\right)^2 - \left(\frac{i_q^* - i_q}{i_n}\right)^2 \tag{10}$$

Next, the method for evaluating an actor's action is the action-value function, which can be described by the Bellman equation:

$$Q^{\mu}(s_t, a_t) = \mathrm{E}\left(r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right) \tag{11}$$

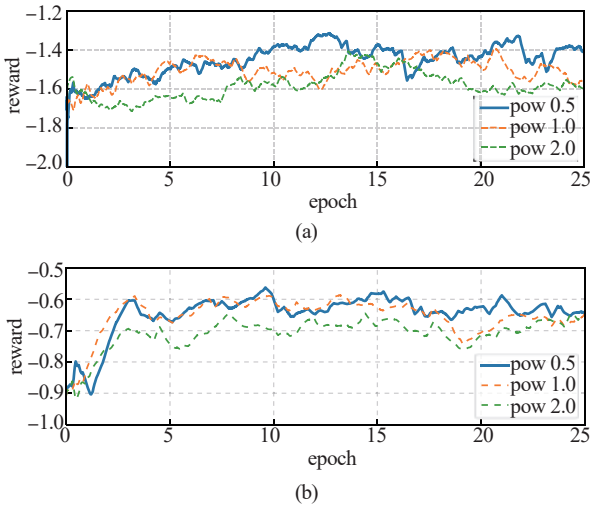

Fig. 4. The actual rewards during training using different power conditons. (a) The rewards of current agent, (b) The rewards of speed agent.

TABLE I
PARAMETERS AND VALUES

Parameters	Values
Stator resistance R	6.0e−2
d component of inductance L_d	1.3e−3
q component of inductance L_q	4.0e−3
Permanent flux ψ_f	125.1e−3
Pole of pairs J	1.324e−3
Moment of inertia R	6.0e−2
Nominal current i_n	195
DC bus voltage u_{dc}	450
Nominal velocity ω_n	2000 × π/30
Sampling time T_s	1e−4

This is a recursive function based on the temporal difference (TD) method, and it includes a discount factor for the future reward. The Q-function can be approximated by a neutral network which is the critic in the agent. During the training, its loss function is given as:

$$\text{Loss} = (r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1})) - Q^{\mu}(s_t, a_t))^2 \quad (12)$$

The second term in the loss function depends on the actor and critic, which are constantly updated during training, and this can introduce instability into the training. Therefore, target functions for the actor (μ') and critic (Q′) are introduced to predict Q-value, and they are set to slowly approach the lasted actor and critic [21]. The calculation of the Q-value requires both state and action, making the actor and critic interdependent and requiring them to work in tandem. As a result of the cascading structure, the gradient can be propagated to the actor, enabling the implement of the gradient ascent algorithm to update the actor's parameters and achieve the maximum reward.

TABLE II
HYPERPARAMETERS USED IN TRAINING

Hyperparameters	Speed agent	Current agent
Number of layers	64/32/32	64/32/32
Activation function of actor	ReLu/ Tanh	ReLu/ Tanh
Activation function of critic	ReLu/ ReLu	ReLu/ ReLu
Learning rate of actor	5e−5	5e−5
Learning rate of critic	3e−5	3e−5
Batch size	128	128
Optimizer	Adam	Adam
Discount factor	0.99	0.99

Note that the environment and reward for the current and the speed controller should be differentiated, while the remaining process is mostly same. To avoid local optima, additional noise is added to actor's output:

$$a_t = \mu(s_t) + N_t \quad (13)$$

It is a crucial aspect of the exploration in RL, and in this paper the noise is used the Ornstein-Uhlenbeck process, which is suitable for physical systems with momentum [21].

B. Change of Reward and Feature Embedding

To ensure the convergence of the algorithm, as depicted in Fig. 3, all variables will be normalized to the range of [–1, 1]. However, normalization will cause the gradient of MSE to become very small as the error gradually converges, which limits the precision of the training. If the exponent of the error in reward is gradually reduced, the gradient of the reward will increase in cases where the error is small. This effect is particularly pronounced when the exponent is less than 1, because the reward becomes more sensitive to fine errors. Thus, this paper made adjustments to the exponent in the rewards. In Fig. 4, the training results of the currents (or speed) reward under three different conditions are demonstrated: when the exponent of error is equal to 2, equal to 1, and equal to 0.5. The system parameter values and the normalization values are given in TABLE I, and fixed hyperparameters used for training are given in TABLE II.

To facilitate comparison, the actual rewards during training in the figure have been uniformly normalized using pow (MAE, 0.1) in Python. It is evident from the figure that reducing the exponent leads to a stable improvement in the actual reward during training. The power is a hyperparameter that can be adjusted based on the specific circumstance. Due to the fact that the exponent becomes too small, it can also affect the gradient when the error is relatively large, and a value of 0.5 is used for the reward in this work.

Data and features are also important factors that influence the training results. In this literature, the features refer to the input vector of the network is expressed in the following form:

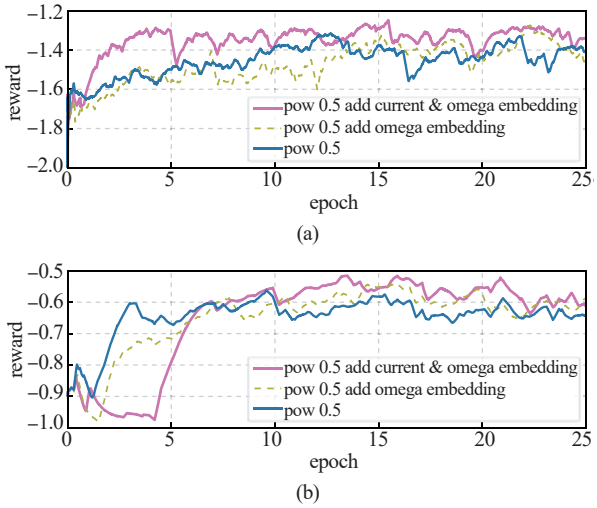

Fig. 5. The actual rewards during training using feature embedding. (a) The rewards of current agent, (b) The rewards of speed agent.

$$v = \left[\frac{i_d}{i_n} \quad \frac{i_q}{i_n} \quad \frac{u_d}{u_n} \quad \frac{u_q}{u_n} \quad \frac{\omega}{\omega_n} \quad \cos\theta \quad \sin\theta \quad \cdots \right. \\ \left. \left(\frac{\omega^* - \omega}{\omega_n} \right) \quad \left(\frac{i_q^* - i_q}{i_n} \right) \right] \tag{14}$$

Taking the example of the speed agent, the feature of speed error in the vector is inevitably important, since the value of the reward is directly determined by it. However, the speed error may only approach 1 during start-up, and for most part, it remains close to 0. This results in a highly imbalanced data distribution, where the feature is effective in distinguishing high and low levels of error, but lacks discriminability when error is low. To some extent, the error features of currents also suffer from a similar problem. Batch normalization can alleviate this limitation, but it relies on the statistical properties of the training data, which can introduce bias during prediction.

To circumvent this barrier, we leverage the feature embedding to better distinguish the magnitude of a certain feature. One approach is to bin the feature and use an LUT to store the embeddings corresponding to the different bins, and adaptively adjust these embeddings during training. The effectiveness of this method depends on the result of binning, and it requires an additional LUT. It is noticeable that, in this work, we use a practical solution by normalizing the same feature differently and concatenating the results into a vector to represent the embedding of this feature. Thus, the embedding of a feature can be express in the following equivalent form:

$$e_v = \left[\left(\frac{v^{(1/l)}}{v_n^{(1/l)}} \right) \quad \left(\frac{v^{(2/l)}}{v_n^{(2/l)}} \right) \quad \cdots \quad \left(\frac{v^{(1/l)}}{v_n^{(1/l)}} \right) \right] \tag{15}$$

where e_v denotes the embedding of feature v, and v can be any feature, such as speed or speed error. The l is a hyperparameter, which is related to the dimension of the embedding.

From (15), it can be seen that small errors will be amplified after normalization. Additionally, the dimension and normalization method can be adjusted based on the results obtained. Fig. 5 shows the actual rewards obtained by the current (or speed) controller during training with features embeddings when l = 4. The blue line in Fig. 5 represents the same results as the blue line in Fig. 4. It can be observed that after adding the current and current error embeddings, the current controller has significantly improved in terms of reward. Similarly, after adding the speed and speed error embeddings, the speed controller also shows prominent enhancement. To illustrate the practical implementation of our modification clearly, the pseudocode for training the controllers is provided in TABLE III.

TABLE III
PSEUDOCODE FOR THE CURRENT/SPEED CONTROLLER

Method: Implementation of the Suggested Controller

Initialize the Q and μ
Initialize the target functions Q', and μ',
for *episode* = 1, M do
  if training current controller, then
    Initialize the Environment 2.
    Neglect the (4) and select reward (10).
  else
    Initialize the Environment 1 and select reward (9).
  for t = 1, T do.
    Select action a_t through (13) and simulate.
    Observe the new state s_{t+1} and calculate the r_t.
    Store(a_t, s_t, r_t, s_{t+1}) in replay buffer.
    Sample a batch of data randomly from replay buffer.
    Update the parameters of Q by minimizing (12).
    Update the parameters of μ by maximizing (11).
    Softly update the target functions Q', and μ'.
    end for.
end for

IV. EVALUATION AND RESULTS

In this section, to verify our theoretical findings, a case study is carried out on a PMSM drive control system, and the functionality of the suggested algorithm will be demonstrated. For a fair comparison, same parameters are set.

The training data are generated by training the actor and critic networks with randomly selected speed setpoints until convergence. To ensure consistency between the training data and the real-world system behavior, a physics-based model of the inverter and PMSM is constructed within a Python environment, along with a discrete-time control system. Specifically, a simulation framework resembling Simulink is developed based on system equations and numerical solvers, allowing the agent to interact with the environment. Practical features such as dead-time effects and digital control delays are also preserved. As a result, the simulation environment produces outputs that match those of the actual physical system.

The speed control agent generates a reference torque based on the system states and the reference speed. Meanwhile, the current control agent outputs a reference voltage according to the system states and the reference torque. Further, the action

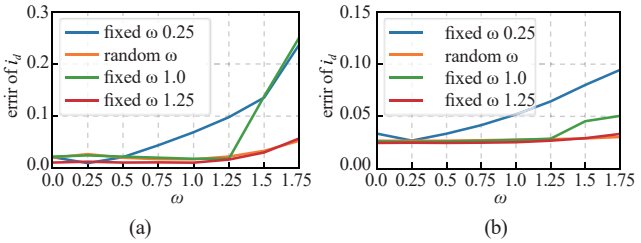

Fig. 6. Performance of the current controller at different speed conditions. (a) Performance of i_d traking, (b) Performance of i_q traking.

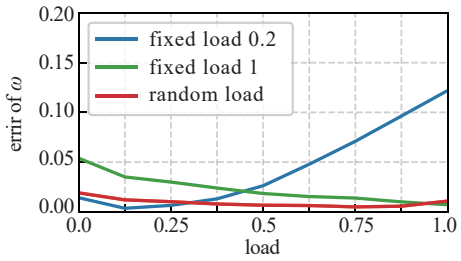

Fig. 7. Performance of the speed controller under different loads.

space of the current control agent is continuous and normalized to the range (–1, 1), based on the allowable voltage range at the inverter output. Similarly, the action space of the torque control agent is continuous and normalized to (–1, 1), according to the torque range at the motor output.

A. Training with Different Speeds and Loads

First, when the current controller is trained, it is important to vary the speed setting in Environment 2 to different values. Although the RL controller has some degree of robustness, in general, using a fixed speed during training can result in significant errors when tested at other speeds. Fig. 6 depicts the performance of the controllers at different speeds during test after being trained at either random range ([0, 1]) or fixed speeds. From Fig. 6, it can be observed that the tracking per formance of i_q (or i_d) is greatly affected when the fixed operating speed is 0.25 in training and the operating speed is 1 in test. This figure indicates that employing random speed during training or chose a larger fixed speed is necessary to maintain desired control performance across the full speed range.

Similarly, it is necessary to use different loads during training process of the speed controller. Fig. 7 also demonstrates that using random loads during training can result in a controller with better performance across different loads.

B. Effectiveness of Feature Embedding

The partially tested performance of the current controller trained using the scheme described above is shown in Fig. 8. As shown in the figure, the model trained using feature embedding outperforms the model without embedding in tracking. This is consistent with the result of reward during training.

Fig. 9 also demonstrates that for the RL speed controller,

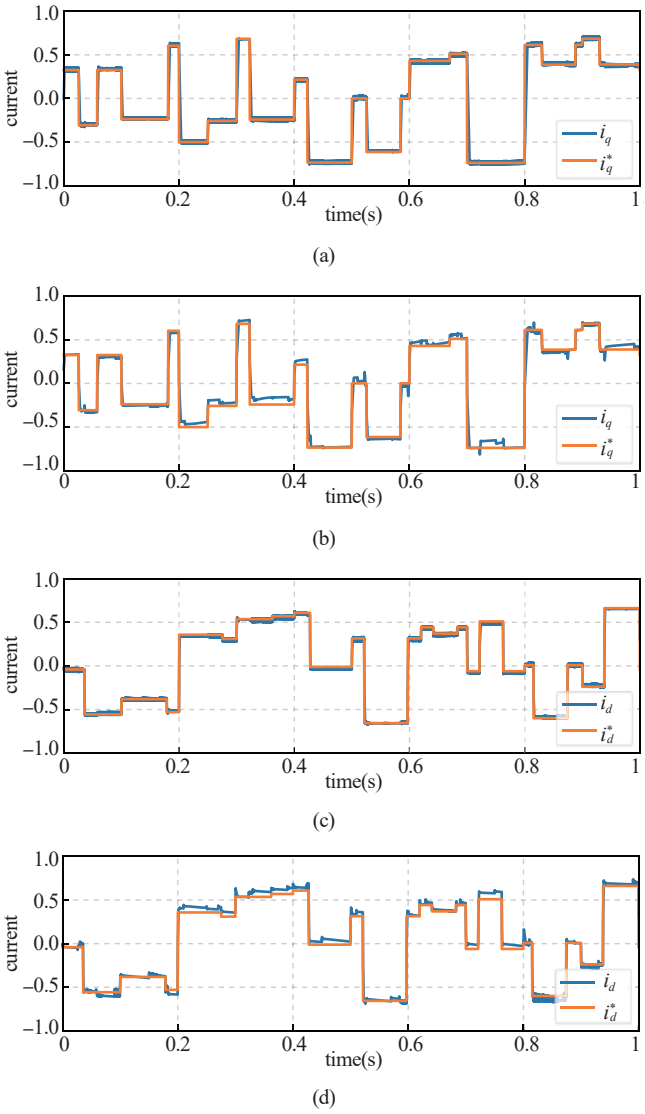

Fig. 8. Performance of the current controller at speed 1.0. (a) i_q traking of the RL controller with embedding, (b) i_q traking of the RL controller without embedding, (c) i_d traking of the RL controller with embedding, (d) i_d traking of the RL controller without embedding.

using embedding results in higher accuracy and less vibration. In Fig. 9, the speed trajectory using a PI controller is also shown. It can be observed that the dynamic performance of the RL controller is superior to that of the implemented traditional model-based controller, and with the help of feature embedding, it can approach the performance of the PI controller in steady-state.

Fig. 10 presents detailed test results, evaluating the performance of these two RL controllers under different operation conditions. This figure not only demonstrates the reliability of the benefits of using embedding, but also highlight two interesting points. In the current test results shown in the figure, it can be seen that the model without embedding experiences an immediate increase in error when the speed exceeds 1, even with random speed range ([0, 1]) during training. This suggests that the generalization ability of the model trained with random speeds still depends on the representation of the features. An-

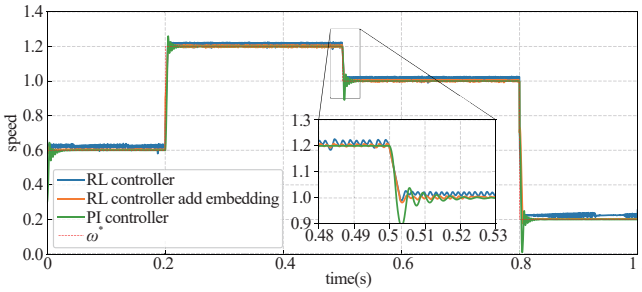

Fig. 9. Performance of the speed controller under load 0.75.

other point that can be observed by comparing D and E is that as long as the speed controller employs the embedding, the use of embedding in the current controller has little effect on speed control performance. This is because for the speed controller, the current controller is only a part of its external environment, which it needs to implicitly learn during training. However, in that sense, if the inner loop cannot accurately track the reference value, it will lead to a discrepancy between actual operating point and operating point from strategy selection module.

Finally, the comparison results also show that the controllers trained by the proposed approach can achieve desired performance to model-based controllers, and have better dynamics, while relying solely on data, and making it suitable for sensitive applications such as transportation. In conclusion, this test illustrates the capabilities of our method to obtain a high-performance under different operation conditions, and our solution works as expected.

V. Conclusions and Future Work

This article demonstrated how to train two cascaded RL controllers in a dual-loop system. By redefining the environments of the two agents and training them sequentially, the current and speed controllers can converge under different operating conditions successfully. Meanwhile, the proposal in this work, which utilized embedding to represent the speed, current and error, can significantly improve the accuracy of the RL controllers when compared to the previous RL controllers. This had been validated in both training and testing. Furthermore, the accuracy and robustness of the RL controller were enhanced by adjusting the reward function and using different operating conditions during training. Finally, the results demonstrated that our development can offer good tracking performance and regulation properties in contrast to two different control approaches, which enable the system to operate as their enhancement, facilitating its quick adoption by the industry.

Future investigations will focus on issues kept out of the scope of this work. First of all, it is expected that the results in this work can be extended to other electric drive systems under cyber attacks, where exploration would be beneficial by addressing an online safety-enhanced deep RL along this study line [22]–[24]. Alternatively, how to design a transferring learning-based long-horizon MPC solution subject to unknown uncertainties is another potential theme that needs further research [25], [26].

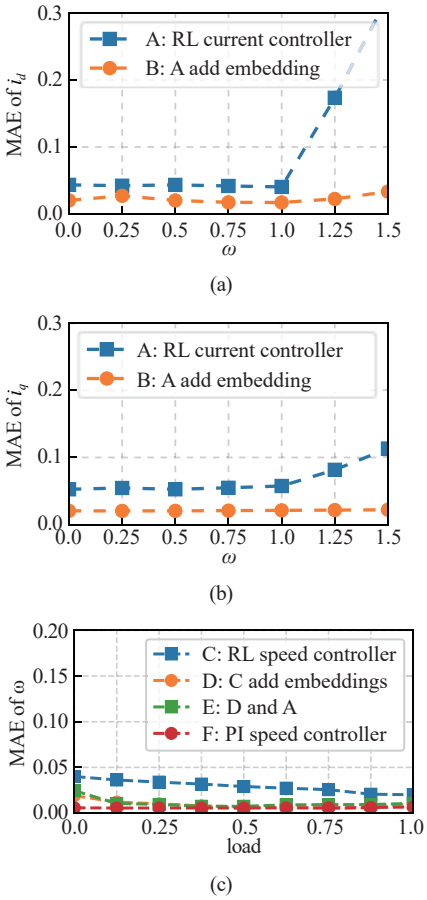

Fig. 10. Performance of the controllers under different conditions. (a) Performance of i_d traking, (b) Performance of i_q traking, (c) Performance of speed traking.

References

[1] J. Rodriguez, C. Gaecia, A. Mora, F. F.-Bahamonde, P. Acuna, M. Novak, Y. Zhang, L. Tarisciotti, S. A. Davari, Z. Zhang et al.,"Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies," in *IEEE Transactions on Power Electronics*, vol. 37, no. 4, pp. 3927–3942, Apr. 2022.

[2] J. Rodriguez, C. Garcia, A. Mora, S. A. Davari, J. Rodas, D. F. Valencia, M. Elmorshedy, F. Wang, K. Zuo, L. Tarisciotti et al., "Latest advances of model predictive control in electrical drives—Part II: Applications and benchmarking with classical control methods," in *IEEE Transactions on Power Electronics*, vol. 37, no. 5, pp. 5047–5061, May 2022.

[3] P. Catalán, Y. Wang, J. Arza, and Z. Chen, "Advanced fault ride-through operation strategy based on model predictive control for high power wind turbine," in *IEEE Transactions on Sustainable Energy*, vol. 15, no. 1, pp. 513–526, Jan. 2024.

[4] T. T. Nguyen, H. N. Tran, T. H. Nguyen, and J. W. Jeon, "Recurrent neural network-based robust adaptive model predictive speed control for PMSM with parameter mismatch," in *IEEE Transactions on Industrial Electronics*, vol. 70, no. 6, pp. 6219–6228, Jun. 2023.

[5] I. González-Prieto, M. J. Duran, A. Gonzalez-Prieto, and J. J. Aciego, "A simple multistep solution for model predictive control in multiphase electric drives," in *IEEE Transactions on Industrial Electronics*, vol. 71, no. 2, pp. 1158–1169, Feb. 2024.

[6] F. Wang, Y. Wei, H. Young, D. Ke, H. Xie, and J. Rodríguez, "Continuous-control-set model-free predictive fundamental current control for PMSM system," in *IEEE Transactions on Power Electronics*, vol. 38, no. 5, pp. 5928–5938, May 2023.

[7] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M.

Norambuena, "Model predictive control for power converters and drives: advances and trends," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, Feb. 2017.

[8] Y. Zhang, J. Jin, and L. Huang, "Model-free predictive current control of PMSM drives based on extended state observer using ultralocal model," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 993–1003, Feb. 2021.

[9] F. Wang and L. He, "FPGA-based predictive speed control for PMSM system using integral sliding-mode disturbance observer," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 972–981, Feb. 2021.

[10] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Combining data-driven and event-driven for online learning predictive control in power converters," in *IEEE Transactions on Power Electronics*, vol. 40, no. 1, pp. 563–573, Jan. 2025.

[11] X. Liu, L. Qiu, J. Rodríguez, K. Wang, Y. Li, and Y. Fang, "Learning-based resilient FCS-MPC for power converters under actuator FDI attacks," in *IEEE Transactions on Power Electronics*, vol. 39, no. 10, pp. 12716–12728, Oct. 2024.

[12] D. Jakobeit, M. Schenke, and O. Wallscheid, "Meta-reinforcement-learning-based current control of permanent magnet synchronous motor drives for a wide range of power classes," in *IEEE Transactions on Power Electronics*, vol. 38, no. 7, pp. 8062–8074, Jul. 2023.

[13] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Event-driven based reinforcement learning predictive controller design for three-phase NPC converters using online approximators," in *IEEE Transactions on Power Electronics*, vol. 40, no. 4, pp. 4914–4926, Apr. 2025.

[14] A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a reinforcement learning environment toolbox for intelligent electric motor control," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 919–928, Mar. 2022.

[15] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Predictive control of voltage source inverter: An online reinforcement learning solution," in *IEEE Transactions on Industrial Electronics*, vol. 71, no. 7, pp. 6591–6600, Jul. 2024.

[16] J. Ye, H. Guo, B. Wang, and X. Zhang, "Deep deterministic policy gradient algorithm based reinforcement learning controller for single-inductor multiple-output DC-DC converter," in *IEEE Transactions on Power Electronics*, vol. 39, no. 4, pp. 4078–4090, Apr. 2024.

[17] M. Schenke, W. Kirchgässner, and O. Wallscheid, "Controller design for electrical drives by deep reinforcement learning: A proof of concept," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4650–4658, Jul. 2020.

[18] M. Schenke and O. Wallscheid, "A deep Q-learning direct torque controller for permanent magnet synchronous motors," in *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 388–400, 2021.

[19] Y. Wang, S. Fang, and J. Hu, "Active disturbance rejection control based on deep reinforcement learning of PMSM for more electric aircraft," in *IEEE Transactions on Power Electronics*, vol. 38, no. 1, pp. 406–416, Jan. 2023.

[20] K. Liu, C. Hou, and W. Hua, "A novel inertia identification method and its application in PI controllers of PMSM drives," in *IEEE Access*, vol. 7, pp. 13445–13454, 2019.

[21] T. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Computer Science*, pp. 1–14, Sept. 2015.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," in *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[23] X. Liu, L. Qiu, Y. Fang, K. Wang, Y. Li, and J. Rodríguez, "Finite control-set learning predictive control for power converters," in *IEEE Transactions on Industrial Electronics*, vol. 71, no. 7, pp. 8190–8196, Jul. 2024.

[24] Y. Wan, Q. Xu, and T. Dragičević, "Safety-enhanced self-learning for optimal power converter control," in *IEEE Transactions on Industrial Electronics*, vol. 71, no. 11, pp. 15229–15234, Nov. 2024.

[25] M. Abu-Ali, F. Berkel, M. Manderla, S. Reimann, R. Kennel, and M. Abdelrahem, "Deep learning-based long-horizon MPC: robust, high performing, and computationally efficient control for PMSM dives," in *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 12486–12501, Oct. 2022.

[26] C. Cui, T. Yang, Y. Dai, C. Zhang, and Q. Xu, "Implementation of transferring reinforcement learning for DC-DC buck converter control via duty ratio mapping," in *IEEE Transactions on Industrial Electronics*, vol. 70, no. 6, pp. 6141–6150, Jun. 2023.

Xing Liu received the Ph.D. degree in Marine Electric Engineering from Dalian Maritime University, Dalian, China, in 2018. From Dec. 2018 to Jan. 2019, he joined the Key Laboratory of Marine Technology and Control Engineering of the Ministry of Communications at Shanghai Maritime University, Shanghai, China. He is currently a Research Fellow with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. He is also currently a Research Fellow with the State Key Laboratory of Power Transmission Equipment Technology, Chongqing University, Chongqing, China. He is also currently a Research Fellow with the State Key Laboratory of High-Speed Maglev Transportation Technology, Qingdao, China. He is also currently a Visiting Scholar in the area of High-Power Converters and Renewable Energy Generation with the Department of Electrical Engineering, Tsinghua University, Beijing, China. He has authored more than 35 publications, all of which are the first authors, including papers in prestigious journal, such as *IEEE Transactions on Industrial Electronics, IEEE Transactions on Power Electronics, IEEE Transactions on Transportation Electrification, IEEE Transactions on Control Systems Technology, IEEE Transactions on Mechatronics, IEEE Journal of Emerging and Selected Topics in Power Electronics, ISA Transactions,* and *International Journal of Electrical Power & Energy Systems*, et al.

His main research interests include finite control-set model predictive control and data-driven predictive control of power converter control systems with applications in power grids, microgrids, and power electronics converters, cybersecurity of power-electronic-intensive electrical distributions systems and microgrids, and applications of artificial intelligence in industrial power electronics and systems.

Dengyin Jiang received the the Ph.D. degree in Control Theory and Control Engineering from Shanghai Jiao Tong University in 2016. He is currently a Lecturer with the School of Electrical Engineering, Shanghai Dianji University, Shanghai, China. His current research interests include intelligent control of electrical machines, transportation electrification, and secure operation of microgrids.

Chenghao Liu received the B.S. degree in Electrical Engineering from Nanjing Institute of Technology, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree in electrical engineering with Zhejiang University, Hangzhou, China. His research interests include data-driven model predictive control and applications of artificial intelligence in industrial power electronics.