Reinforcement Learning for Intelligent Control of AC Machine Drives: A Review

Nabil Farah
School of Electrical and Data Engineering
University of Technology Sydney
NSW, Australia
nabil.farah@student.uts.edu.au

Gang Lei
School of Electrical and Data Engineering
University of Technology Sydney
NSW, Australia
gang.lei@uts.edu.au

Jianguo Zhu
School of Electrical and Information Engineering
University of Sydney
NSW, Australia
jianguo.zhu-@sydney.edu.au

Youguang Guo
School of Electrical and Data Engineering
University of Technology Sydney
NSW, Australia
youguang.guo-1@uts.edu.au

***Abstract*— Permanent magnet synchronous motors (PMSMs) are widely used in various industrial applications due to their high efficiency, compact size, and precise control capabilities. However, traditional control techniques often struggle to handle the nonlinearities and uncertainties associated with PMSM drives. Reinforcement learning (RL) based control approaches have offered a promising solution to address these challenges. This article reviews RL-based control of PMSM drives, delving into fundamental concepts, machine learning types, and RL frameworks. Challenges, drawbacks, and future directions for enhancing RL-based control methods are also discussed.**

Keywords—PMSM, machine learning, RL, uncertainties, challenges.

I. INTRODUCTION

Permanent magnet synchronous motors (PMSMs) are widely used in various industrial applications due to their high efficiency, compact size, and precise control capabilities. Ensuring precise and efficient control of PMSM drives is essential to optimise system performance and minimise energy consumption. However, the inherent complexities and uncertainties in PMSM dynamics present significant challenges for traditional control methods, leading to suboptimal performance in varying operating conditions[1].

Conventional control techniques, such as field-oriented control (FOC), direct torque control (DTC), model predictive control (MPC) and other deterministic approaches, often struggle to handle the nonlinearities and uncertainties associated with PMSM drives[2]. These uncertainties arise due to variations in motor parameters, temperature changes, external disturbances, and uncertainties in the load. As a result, the performance of traditional control methods can be limited, leading to reduced accuracy and robustness[1].

In recent years, reinforcement learning (RL) based control approaches have offered a promising solution to address the challenges posed by uncertainties in PMSM drives[3]. RL is a data-driven control technique that does not require explicit knowledge of the motor parameters. Instead, RL agents learn optimal control policies through interaction with the environment, making them robust to uncertainties and disturbances[4]. RL-based control methods can adapt to changes in motor parameters and optimise control performance in real-time. Furthermore, RL-based control methods offer computational efficiency compared to some other data-driven control techniques, such as model-free control. RL agents can learn offline optimised policies, which eliminates the need for online optimization during operation[5].

RL can be employed to enhance the performance of standard PMSM control strategies (i.e., FOC, DTC, and MPC). For instance, [6] utilized RL to obtain the weight coefficients of an improved MPC for PMSM drives, and [7, 8] implemented deep RL to optimize the parameters of active disturbance rejection control of PMSM. Furthermore, RL can replace the standard control methods of PMSM drives entirely. RL-based current control [3] and torque control [9] of PMSM drives were implemented by training a deep Q-learning network to learn optimal controllers. These learned-based controllers were then deployed to a real-world drive system and demonstrated comparable performance to the standard controllers[4]. RL-based speed control was trained to achieve optimal speed tracking and replace standard speed control [10].

However, there are some challenges that must be addressed when applying RL to PMSM drives. One major challenge is the complexity of the control problem, which involves balancing conflicting objectives such as torque regulation, speed control, and energy efficiency. Additionally, the PMSM drive system has many interconnected components that must be considered, including the motor, power electronics, and control algorithms. Another challenge is the difficulty of training RL algorithms in real-time. Training an RL algorithm requires a large amount of data, which can be time-consuming and computationally intensive. Additionally, the training process can be sensitive to the choice of hyperparameters, such as learning rate and discount factor, which can affect the stability and convergence of the algorithm[11].

This article reviews RL-based control of PMSM drives, delving into fundamental concepts, machine learning types, and RL frameworks. The formulation and RL application in PMSM control are examined. Challenges, drawbacks, and future directions for enhancing RL-based control methods are also discussed. The paper aims to enhance the understanding of RL's potential in optimizing PMSM drive control and serves as a benchmark for future advancements in RL-based control techniques.

The rest of the paper is organized as follows: Section II discusses the fundamental concept of RL; Section III presents RL based PMSM drives. Sections IV and V discuss the challenges and future perspective of RL based PMSM drives. Section VI presents the conclusion.

II. Fundamentals of RL

Machine learning is a branch of artificial intelligence that involves the use of algorithms and statistical models to enable computer systems to learn from data and improve their performance on a specific task over time[12]. It has brought significant advancements to fields such as natural language processing, computer vision, and robotics, enhancing the capabilities of machines to perform tasks that once seemed beyond their reach[13]. Among the myriad of applications, machine learning finds a compelling use case in controlling AC motor drives, where its versatile types can be harnessed to optimize motor performance, energy efficiency, and overall system operation[14]. There are different types of machine learning, including supervised, unsupervised, and reinforcement learning, each with its unique characteristics and applications[15].

Supervised Learning: Supervised learning is one of the fundamental types of machine learning, where the algorithm learns from labelled data to make predictions or decisions [16]. In contrast, unsupervised learning involves training the algorithm on unlabelled data to discover patterns and relationships within the dataset. For AC motor drives, unsupervised learning can be used for clustering similar motor operating states or identifying anomalies in motor behaviour. This aids in predictive maintenance, as it allows the system to detect potential faults or irregularities before they escalate into major issues.

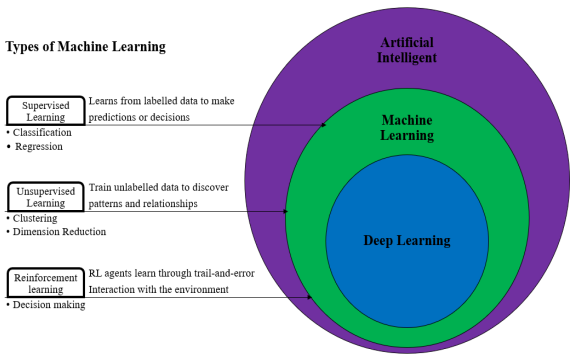

Fig. 1. Types of Machine Learning

Reinforcement learning (RL) is an interactive learning approach where an agent learns to take actions in an environment to maximize a cumulative reward. When applied to AC motor drives, RL can optimize control strategies by exploring various control actions, observing the motor's responses, and adjusting its decisions to achieve specific objectives, such as energy efficiency or torque regulation[9, 17]. RL differs from supervised and unsupervised learning because it operates in a dynamic environment and learns from collected experiences rather than static datasets. During training, RL agents gather data through trial-and-error interactions, eliminating the need for data collection, preprocessing, and labelling. RL methods can autonomously learn behaviours without human supervision, making them adaptable to complex environments. Due to these advantages, RL methods are preferred for AC drive control [14, 18].

A. RL Workflow

The learning process aims to maximize a predetermined cumulative reward through decisions based on input signals of observations and rewards. Observations are a set of signals that define the process, while rewards measure the success of actions. Actions are controlled process quantities, and observations are measured signals, their rate of change, and associated errors visible to the agent[19]. Fig. 2 shows the RL process's general block diagram, which includes an agent, environment, action, observations, and rewards. At each time step k, the agent executes an action, receives observations O_k and rewards R_k. The environment receives an action A_k , and emits observation O_{k+1} and scalar reward R_{k+1}. RL is based on the reward hypothesis, and the agent's job is to maximize g_k which is the discounted future rewards:

$$g_k = \sum_{i=0}^{\infty} \gamma^i R_{k+i+1} \tag{1}$$

Another part of the RL is the history and state. History is the sequence of observations, actions, and rewards.

$$H_k = O_1, R_1, A_1, \ldots, A_{k-1}, O_k, R_k \tag{2}$$

The state is the information used to determine what happens next; it is a function of history: $S_t = f(H_t)$.

RL based control workflow involves creating an environment, defining rewards, creating the agent, comprising the policy and the RL training algorithm, and deploying the trained policy as a standalone decision-making system[20].

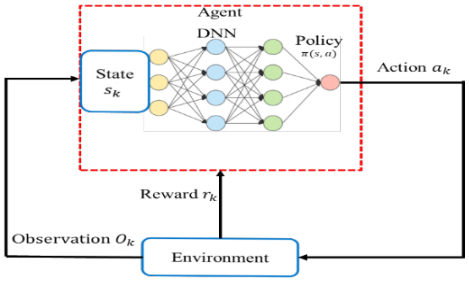

Fig. 2. General block diagram RL process.

B. RL Agent

An RL agent consists of essential components that allow it to interact with and learn from the environment. The policy, which determines the agent's behaviour or strategy, can be deterministic or stochastic. In a deterministic policy, the agent directly chooses an action

based on the state, while in a stochastic policy, actions are selected probabilistically, considering the current state.

Secondly, the value function assesses the potential future rewards that an agent can expect to accumulate. It serves as a measure to evaluate the goodness or badness of states and helps the agent make decisions about which actions to take in different situations. There are two primary types of value functions: the state-value function ($v_\pi\ (s)$), which represents the expected return starting from a specific state and following a policy π, and the action-value function ($q_\pi\ (s_k, u_k)$), which represents the expected return when taking a particular action (u_k) in a particular state (s_k) and then following the policy π thereafter.

Lastly, the model is an essential component that represents the agent's understanding or prediction of how the environment behaves. It acts as a simulation of the environment and can predict the next state and the immediate reward when given the current state and an action. Furthermore, RL agents can be categorized based on the presence and absence of policy and value function components. These categories encompass value-based, policy-based, and Actor-Critic agents, each serving a unique role in the RL landscape.

C. RL Formulation

RL is a fundamental machine learning approach based on a Markov decision process (MDP) represented by a tuple (S, A, P, r, γ). In this setup, the environment defines the state space (S), while the agent has the action space (A). The agent interacts with the environment to update its policy,π, which maps states to actions. At each step, the agent selects an action ($a_k \in A$) based on its policy π, and the environment generates the next state (s_{k+1}) using a transition probability function (P). The environment also provides the agent with a reward (r) as feedback. This process continues until the agent finds the optimal policy (π^*), that is expressed as follows:

$$\pi^* \in \arg\max_{\pi} J(\pi) = E\sum_{k=1}^{\infty} \gamma^k\ R\ (s_k, a_k) \qquad (3)$$

where γ represents the discounting factor, and $J(\pi)$ denotes the infinite horizon discounted reward.

The optimal policy ensures that the agent accumulates the maximum possible reward from the environment. To estimate how good a state is, either a state value function $V(s)$ or a state-action value function $Q(s, a)$ is used. When using the V function, value iteration aims at finding the optimal values V^* based on Bellman optimality equation:

$$V^*(s) = \max_a \left[r(s,a) + \gamma \sum_{y \in S} P(s,a,y) V*(y) \right] \quad (4)$$

where $r(s, a)$ is the reward obtained by taking action a in state s,$P(s, a, y)$ is the probability of reaching state y when taking action, a in state s.

The state-action value function $Q_\pi(s, a)$ defines the value of being in state s, taking action a then following policy π. The Bellman optimality equation for Q^* is:

$$Q^*(s,a) = r(s,a) + \gamma \sum_y P(s,a,y) \max_{a'} Q^*(y,a') \quad (5)$$

The policy iteration algorithm is more complicated than value iteration. Given a MDP and a policy π, policy iteration iterates the following steps:

- Evaluate: compute V or Q based on the policy π.
- Improve: compute a better policy based on V or Q.

This process is repeated until convergence. When using V, $V^\pi\ (s)$ is the expected return when starting from state s and following policy π is processed based on the Bellman optimality equation for deterministic policies:

$$V^\pi(s) = r\big(s, \pi(s)\big) + \gamma \sum_{y \in S} P(s, \pi(s), y)\, V^\pi(y) \quad (6)$$

When using Q, the Q equation with policy π becomes:

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{y\,\in S} P(s,a,y) Q^\pi\big(y, \pi(y)\big) \quad (7)$$

III. RL Based PMSM Drives

A. State of Art

RL has emerged as a powerful paradigm for the control PMSM, offering significant advancements in the field of motor control. PMSMs are widely utilized in various applications, ranging from electric vehicles to industrial robotics, where precise and efficient control is essential. Traditional control methods for PMSMs, such as PI controllers, have limitations in handling complex and nonlinear motor dynamics, making RL an attractive alternative[21].

RL leverages its ability to learn optimal control policies through interaction with the system. This interaction typically involves an agent, which represents the controller, taking actions in an environment, represented by the PMSM drive, and receiving feedback in the form of rewards based on the achieved performance. Over time, the agent learns to make decisions that maximize cumulative rewards, effectively optimizing the motor control strategy[17].

RL can be used to tune or enhance the performance of a conventional PMSM drives (i.e., FOC,DTC,MPC) strategy such as [6] which implement RL to tune the weighting factor of MPC. RL also can be used to replace these strategies such as RL-based current control[3] , torque control [9] , and speed control [10] of PMSM drives. To develop RL-based control of PMSM, environment, observation, rewards, action, and agent must be defined.

The effectiveness of RL-based controllers heavily relies on the quality and quantity of training data. Standard RL controllers aim to learn optimal policies for specific task conditions and parameter sets, making them vulnerable to poor performance or instability when faced with new conditions or parameter mismatches[22]. This challenge of adapting to new scenarios is addressed through meta-RL[5], which efficiently leverages prior experience on similar tasks. In this approach, a dataset of diverse motor parameters is utilized to model each motor's environment as a partially observable Markov decision process (POMDP),

where additional contextual variables capture momentary environmental information, to improve adaptability.

RL-based PMSM drives comprise key components, including the environment, observation, action, and reward. In this context, the environment is embodied by the PMSM system itself, serving as the dynamic framework within which the control operates. Observations, on the other hand, encompass the data and information collected from the system, while actions denote the decisions and control inputs made by the RL agent. The reward signal guides the learning process, providing feedback to the agent based on its actions in pursuit of optimal motor control. This interconnected framework forms the basis for developing adaptive and efficient PMSM drive controllers through reinforcement learning techniques[17].

B. RL Environment

The RL environment serves as a platform for the interaction between an RL agent and its surroundings. In the case of PMSMs, the RL environment represents the operational context in which the motor operates. In general, RL-based current control of PMSM drives is trained offline using a simulation model represented by mathematical equations that describes the dynamics of the PMSM:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \tag{8}$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega L_d i_d + \omega \Psi_{pm} \tag{9}$$

$$\frac{di_d}{dt} = -\frac{R_s}{L_d} i_d + \frac{L_q}{L_d} \omega i_q + \frac{1}{L_d} v_d \tag{10}$$

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q - \frac{L_d}{L_q} \omega i_d - \frac{\Psi_{pm}}{L_q} \omega + \frac{1}{L_q} v_q \tag{11}$$

C. Observation,Rewards and Action

The selection of appropriate observations, rewards, and actions is a crucial factor in determining the effectiveness of the RL-based controller. For current control of PMSM drives, standard observations are measured and reference *dq* currents, measured and reference speed, measured position, and *dq* voltages, expressed as:

$$o^k = \left[\omega^*, \omega, \theta, i_d^*, i_d^k, i_q^*, i_q^k, v_d^{k-1}, v_q^{k-1}\right]^T \tag{12}$$

Rewards are an essential part of the RL learning process; it tells the agent how good or bad the selected action is. Therefore, rewards must be appropriately calculated to help the agent learn an optimal policy. In current control, the objective is to minimize the current error, and the quadric objective function can be written as:

$$g^k = \left(i_d^* - i_d^k\right)^2 + \left(i_q^* - i_q^k\right)^2 \tag{13}$$

The objective function g^k, along with constraint violation penalty are employed as a reward signal for RL agent as follows:

$$r^k = -(w_1 g^k + w_2 P^k) \tag{14}$$

where $w_1 and\ w_2$ are the reward gains, P^k is a penalty term to ensure safe operation and discourage overcurrent region during the training by penalizing the agent when the measured current $i_s^k = \sqrt{(i_d^k)^2 + \left(i_q^k\right)^2}$ exceeds the nominal current i_n, and can be calculated as the following:

$$P^k = \begin{cases} (i_n - i_s^k)^2 \ , i_s^k > i_n \\ 0 \ , otherwise \end{cases} \tag{15}$$

RL agents can have either discrete or continuous action spaces. Q-learning and deep Q-network (DQN) agents have a discrete action space, while deep deterministic policy gradient (DDPG) and twin-delayed deep deterministic policy gradient (TD3) agents have a continuous action space. PMSM drive based RL with discrete actions, the power electronic converter's nature is considered, and the action can be one of the possible switching vectors, s_j. With a two-level three-phase inverter, eight possible vectors can be applied:

$$a^k = s_j \in [0,1,2 \dots \dots 7] \tag{16}$$

For continuous action space, the agent actions can be the d- and q-axes voltage components:

$$a^k = \left[v_d^k \quad v_q^k\right]^T \tag{17}$$

IV. Challenges and Limitations of RL

While RL has shown promise in improving the control of PMSM drives, there are several drawbacks and disadvantages to existing RL-based methods, including:

1. High computational requirements: RL algorithms can require high computational resources, particularly when using deep neural networks to approximate the Q-function or policy. This can make RL-based control methods impractical for real-time with limited processing power or strict latency requirements[23].
2. Difficulty in generalizing to different operating conditions: RL agents are trained on specific operating conditions and may not generalize well to new or unexpected conditions. This can lead to poor performance or instability in the controlled system. Ensuring that the RL agent is robust and able to handle a range of operating conditions is an ongoing challenge[5].
3. Sensitivity to hyperparameters: RL algorithms often have many hyperparameters that need to be carefully tuned to achieve good performance. This can be time-consuming and requires expertise in RL methods, making it difficult for non-experts to use RL in practice[24].
4. Need for extensive training data: RL algorithms require large amounts of training data to learn effective control policies, particularly when using deep neural networks. Collecting and annotating this data can be expensive and time-consuming[25].

V. Future Prospective

The future direction of RL is likely to involve advancements in areas such as Deep RL, Multi-Agent RL, transfer learning, and explainable RL. These advancements will enable agents to learn more complex behaviors, make decisions based on more complex inputs, and improve trust and transparency in their use[26]. In the context of PMSM drives, RL-based control methods are ongoing research area with some potential future directions:

1. Online Learning: Integrating online learning into RL-based PMSM drives can enable the system to continuously adapt to changing conditions and variations in the motor's characteristics. Online learning algorithms would allow the control policy to be updated in real-time, ensuring better performance and responsiveness to dynamic environments[27-29].

2. Transfer Learning: Applying transfer learning to RL-based PMSM drives can enhance their efficiency and reduce the training time for new control tasks. By leveraging knowledge learned from previous motor control tasks, the agent can start with a more informed policy and fine-tune it for new drives, leading to faster convergence and improved control performance[30].
3. Multi-Task Learning: RL-based PMSM drives can benefit from multi-task learning, where the agent learns to control multiple motors with different characteristics simultaneously. This approach can lead to better generalization and resource utilization, as knowledge acquired from one task can be leveraged to improve the performance in the related tasks [31].
4. Explainable Reinforcement Learning: explainable RL is an emerging focus in the development of RL-based PMSM drives. It addresses the need for transparency and interpretability in the decision-making process of these systems. By making the actions and choices of RL-controlled PMSM drives more understandable, it enhances safety, trust, and compliance with regulatory standards. It also aids in diagnosing issues, improves human-robot collaboration, and facilitates knowledge transfer, ensuring that these advanced control systems are not only efficient but also comprehensible and reliable in practical applications[32].
5. Hybrid Control Strategies: Combining RL-based approaches with traditional control methods can potentially address some of the drawbacks of pure RL techniques. Hybrid strategies can offer stability guarantees and better system understanding, while RL can handle adaptability and optimization in non-linear or uncertain motor control scenarios[33].

VI. Conclusion

RL-based control methods offer a promising solution to address the challenges posed by uncertainties in PMSM drives. RL agents can learn optimal control policies through interaction with the environment, making them robust to uncertainties and disturbances. However, there are challenges that must be addressed when applying RL to PMSM drives, such as the complexity of the control problem and the difficulty of training RL algorithms in real-time. Future research should focus on enhancing RL-based control methods to optimize PMSM drive control and improve system performance.

References

[1] J. Yang, W. H. Chen, S. Li, L. Guo, and Y. Yan, "Disturbance/Uncertainty Estimation and Attenuation Techniques in PMSM Drives—A Survey," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 4, pp. 3273-3285, 2017, doi: 10.1109/TIE.2016.2583412.

[2] F. Wang, Z. Zhang, X. Mei, J. Rodríguez, and R. Kennel, "Advanced Control Strategies of Induction Machine: Field Oriented Control, Direct Torque Control and Model Predictive Control," *Energies,* vol. 11, no. 1, p. 120, 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/1/120.

[3] M. Schenke, W. Kirchgässner, and O. Wallscheid, "Controller Design for Electrical Drives by Deep Reinforcement Learning: A Proof of Concept," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 7, pp. 4650-4658, 2020, doi: 10.1109/TII.2019.2948387.

[4] G. Book *et al.*, "Transferring Online Reinforcement Learning for Electric Motor Control From Simulation to Real-World Experiments," *IEEE Open Journal of Power Electronics,* vol. 2, pp. 187-201, 2021, doi: 10.1109/OJPEL.2021.3065877.

[5] 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," *IEEE Transactions on Power Electronics,* vol. 38, no. 7, pp. 8062-8074, 2023, doi: 10.1109/tpel.2023.3256424.

[6] J. Gao, J. Zhang, M. Fan, Z. Peng, Q. Chen, and H. Zhang, "Model Predictive Control of Permanent Magnet Synchronous Motor Based on State Transition Constraint Method," *Mathematical Problems in Engineering,* vol. 2021, p. 3171417, 2021/11/25 2021, doi: 10.1155/2021/3171417.

[7] Y. Wang, S. Fang, J. Hu, and D. Huang, "Multiscenarios Parameter Optimization Method for Active Disturbance Rejection Control of PMSM Based on Deep Reinforcement Learning," *IEEE Transactions on Industrial Electronics,* vol. 70, no. 11, pp. 10957-10968, 2023, doi: 10.1109/tie.2022.3225829.

[8] Y. Wang, S. Fang, and J. Hu, "Active Disturbance Rejection Control Based on Deep Reinforcement Learning of PMSM for More Electric Aircraft," *IEEE Transactions on Power Electronics,* vol. 38, no. 1, pp. 406-416, 2023, doi: 10.1109/tpel.2022.3206089.

[9] M. Schenke and O. Wallscheid, "A Deep Q-Learning Direct Torque Controller for Permanent Magnet Synchronous Motors," *IEEE Open Journal of the Industrial Electronics Society,* vol. 2, pp. 388-400, 2021, doi: 10.1109/OJIES.2021.3075521.

[10] J. Zhao, C. Yang, W. Gao, and L. Zhou, "Reinforcement Learning and Optimal Control of PMSM Speed Servo System," *IEEE Transactions on Industrial Electronics,* vol. 70, no. 8, pp. 8305-8313, 2023, doi: 10.1109/tie.2022.3220886.

[11] K. Kudelina, T. Vaimann, B. Asad, A. Rassõlkin, A. Kallaste, and G. Demidova, "Trends and Challenges in Intelligent Condition Monitoring of Electrical Machines Using Machine Learning," *Applied Sciences,* vol. 11, no. 6, p. 2761, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/6/2761.

[12] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science,* vol. 349, no. 6245, pp. 255-260, 2015, doi: doi:10.1126/science.aaa8415.

[13] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics,* vol. 16, no. 6, pp. 321-332, 2015/06/01 2015, doi: 10.1038/nrg3920.

[14] S. Zhang, O. Wallscheid, and M. Porrmann, "Machine Learning for the Control and Monitoring of Electric Machine Drives: Advances and Trends," *IEEE Open Journal of Industry Applications,* vol. 4, pp. 188-214, 2023, doi: 10.1109/OJIA.2023.3284717.

[15] E. F. Morales and H. J. Escalante, "Chapter 6 - A brief introduction to supervised, unsupervised, and reinforcement learning," in *Biosignal Processing and Classification Using Computational Learning and Intelligence*, A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, and O. Mendoza-Montoya Eds.: Academic Press, 2022, pp. 111-129.

[16] Y. Gao, B. Cheong, S. Bozhko, P. Wheeler, C. Gerada, and T. Yang, "Surrogate role of machine learning in motor-drive optimization for more-electric aircraft applications," *Chinese Journal of Aeronautics,* vol. 36, no. 2, pp. 213-228, 2023/02/01/ 2023, doi: https://doi.org/10.1016/j.cja.2022.08.011.

[17] A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a Reinforcement Learning Environment Toolbox for Intelligent Electric Motor Control," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 33, no. 3, pp. 919-928, 2022, doi: 10.1109/TNNLS.2020.3029573.

[18] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[19] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization,* vol. 12, no. 3, p. 729, 2012.

[20] MathWorks. "Reinforcement Learning - MATLAB & Simulink " MathWorks. (accessed Jul.27, 2023).
[21] F. Yin, X. Yuan, Z. Ma, and X. Xu, "Vector Control of PMSM Using TD3 Reinforcement Learning Algorithm," *Algorithms,* vol. 16, no. 9, p. 404, 2023.
[22] A. Ez-zizi, S. Farrell, D. Leslie, G. Malhotra, and C. J. H. Ludwig, "Reinforcement Learning Under Uncertainty: Expected Versus Unexpected Uncertainty and State Versus Reward Uncertainty," *Computational Brain & Behavior,* 2023/03/20 2023, doi: 10.1007/s42113-022-00165-y.
[23] R. Pina, H. Tibebu, J. Hook, V. De Silva, and A. Kondoz, "Overcoming Challenges of Applying Reinforcement Learning for Intelligent Vehicle Control," (in eng), *Sensors (Basel),* vol. 21, no. 23, Nov 25 2021, doi: 10.3390/s21237829.
[24] F. M. Talaat and S. A. Gamel, "RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing,* vol. 14, no. 10, pp. 13349-13359, 2022, doi: 10.1007/s12652-022-03788-y.
[25] T. Zhao *et al.*, "A model-based reinforcement learning method based on conditional generative adversarial networks," *Pattern Recognition Letters,* vol. 152, pp. 18-25, 2021/12/01/ 2021, doi: https://doi.org/10.1016/j.patrec.2021.08.019.
[26] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data,* vol. 8, no. 1, p. 53, 2021/03/31 2021, doi: 10.1186/s40537-021-00444-8.
[27] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing,* vol. 459, pp. 249-289, 2021.
[28] J. Schrittwieser, T. Hubert, A. Mandhane, M. Barekatain, I. Antonoglou, and D. Silver, "Online and offline reinforcement learning by planning with a learned model," *Advances in Neural Information Processing Systems,* vol. 34, pp. 27580-27591, 2021.
[29] C. Jiang, X. Li, J.-R. Lin, M. Liu, and Z. Ma, "Adaptive control of resource flow to optimize construction work and cash flow via online deep reinforcement learning," *Automation in Construction,* vol. 150, p. 104817, 2023.
[30] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2023.
[31] G. Cheng, L. Dong, W. Cai, and C. Sun, "Multi-Task Reinforcement Learning With Attention-Based Mixture of Experts," *IEEE Robotics and Automation Letters,* 2023.
[32] R. Dazeley, P. Vamplew, and F. Cruz, "Explainable reinforcement learning for broad-xai: a conceptual framework and survey," *Neural Computing and Applications,* pp. 1-24, 2023.
[33] D. Zhuang, V. J. Gan, Z. D. Tekler, A. Chong, S. Tian, and X. Shi, "Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning," *Applied Energy,* vol. 338, p. 120936, 2023.