applied sciences

MDPI

Article

Synergizing Transfer Learning and Multi-Agent Systems for Thermal Parametrization in Induction Traction Motors

Fozia Mehboob [1], Anas Fattouh [1,*] and Smruti Sahoo [2]

[1] School of Innovation, Design and Technology (IDT), Mälardalen University (MDU), 631 05 Eskilstuna, Sweden; fozia.mehboob@mdu.se
[2] Alstom, 721 36 Västerås, Sweden; smrutirekha.sahoo@alstomgroup.com
* Correspondence: anas.fattouh@mdu.se

Abstract: Maintaining optimal temperatures in the critical parts of an induction traction motor is crucial for railway propulsion systems. A reduced-order lumped-parameter thermal network (LPTN) model enables computably inexpensive, accurate temperature estimation; however, it requires empirically based parameter estimation exercises. The calibration process is typically performed in labs in a controlled experimental setting, which is associated with a lot of supervised human efforts. However, the exploration of machine learning (ML) techniques in varied domains has enabled the model parameterization in the drive system outside the laboratory settings. This paper presents an innovative use of a multi-agent reinforcement learning (MARL) approach for the parametrization of an LPTN model. First, a set of reinforcement learning agents are trained to estimate the optimized thermal parameters using the simulated data in several driving cycles (DCs). The selection of a reinforcement learning agent and the level of neurons in the RL model is made based on variability of the driving cycle data. Furthermore, transfer learning is performed on a new driving cycle data collected on the measurement setup. Statistical analysis and clustering techniques are proposed for the selection of an RL agent that has been pre-trained on the historical data. It is established that by synergizing within reinforcement learning techniques, it is possible to refine and adjust the RL learning models to effectively capture the complexities of thermal dynamics. The proposed MARL framework shows its capability to accurately reflect the motor's thermal behavior under various driving conditions. The transfer learning usage in the proposed approach could yield significant improvement in the accuracy of temperature prediction in the new driving cycles data. This approach is proposed with the aim of developing more adaptive and efficient thermal management strategies for railway propulsion systems.

Keywords: reinforcement learning; transfer learning; railway propulsion system; induction motor; thermal model; optimization

check for updates

Citation: Mehboob, F.; Fattouh, A.; Sahoo, S. Synergizing Transfer Learning and Multi-Agent Systems for Thermal Parametrization in Induction Traction Motors. *Appl. Sci.* **2024**, *14*, 4455. https://doi.org/10.3390/app14114455

Academic Editor: Frede Blaabjerg

Received: 29 February 2024
Revised: 19 May 2024
Accepted: 21 May 2024
Published: 23 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction and Related Work

Induction motors are widely preferred in railway propulsion system applications due to their mechanical robustness, good control properties, high overload capacities, and affordability [1,2]. However, maintaining optimum temperatures in the critical parts of these traction motors is the key to maximizing their efficiencies and also to maintaining a safe operational margin. The maximization of motor utilization in terms of generating maximum torque and power can lead to a high current load, causing more heat load in the windings. This thermal overloading phenomenon increases the possibility of motor failure if the temperature exceeds the insulation-withstanding capacity. Furthermore, dynamically changing the ambient operating conditions and load demand in a traction drive system can cause the motors' performance to vary nonlinearly due to parasitic effects from the core saturation. Hence, temperature monitoring at the critical parts of the motor is essential to prevent any thermal stress while optimizing the motor's performance. In a drive system

control environment, real-time temperature estimation is fed back to the controller to compensate for electrical signals such as voltage and current [3].

Thermal analyses using computational fluid dynamics (CFD) or the heat-equation-based finite element analysis (FEA) method are quite accurate in estimating temperatures [4] and are performed during the design phase. Nonetheless, these studies demand rigorous modeling effort and are computationally expensive. This makes the applicability of these tools unsuitable for real-time monitoring upfront. Installing contact-based sensors such as thermistors or thermocouples is the simplest means for temperature measurement. While the installation of sensors in the stationary parts of the motor such as stator winding, end-wining, core, etc., are straightforward, for the rotating parts, in addition to the sensors, the use of slip rings or other telemetry means such as Wi-Fi, Bluetooth, or infrared communication is necessary for data transmission. Regardless, the deployment of sensors requires integration efforts and additional costs. It also adds complexity due to their inaccessibility for replacement in the case of failures or detuning. As an alternative, model-based measurement techniques for temperature estimation have been in focus during the past decade [5,6]. With these approaches, the targeted temperatures are estimated from the measurement of temperature-dependent electrical parameters in both offline and online manners. The accuracy of the estimation is dependent on the level of modeling effort in capturing the nonlinear, electro-thermal motor dynamics. This includes magnetic/core saturation, cross-saturation effects, hysteresis, eddy current effects, temperature-dependent material properties variation, etc. These methods are also invasive and create a disturbance during normal operation [7].

An alternative direct-temperature estimation technique is using the lumped-parameter thermal network model [8]. An LPTN model is a simplification of the motor physical model that can be represented in a thermal equivalent circuit diagram. The LPTN model comprises several nodes that represent each part of the motor and thermal parameters that represent the heat conduction, radiation, and convection phenomena between different parts, i.e., the nodes. The heat transfer values are represented as resistances and the thermal masses as capacitances in the LPTN. Thermal parameter values are calculated using heat transfer theory-based analytical equations and motor geometry and material information data. The other essential input to the LPTN models for calculating the temperature are the electro-mechanical power conversion losses, such as winding losses, core losses, and mechanical and windage losses. Equation (1) represents the differential equation that represents the heat balance for each node in a thermal equivalent circuit diagram. Temperature estimation is made by solving a set of equations representing all nodes, using numerical techniques.

$$C_i \frac{dT_i}{dt} = \lambda_i^{cond} + \lambda_i^{conv} + \lambda_i^{rad} + P_i \quad (1)$$

where C_i is the thermal capacitance of node i, T_i is the temperature of node i, λ_i^{cond} is the conductance thermal resistance between node i and its environment, λ_i^{conv} is the convection thermal resistance between node i and its environment, λ_i^{rad} is the radiation thermal resistance between node i and its environment, and P_i is the heat source of node i.

The accuracy of an LPTN-model temperature estimation depends on the number of chosen nodes in the network. Typically, LPTN models are categorized as white box (100–200 nodes) and grey box (2–15 nodes) models. In the white box LPTN-model approach, the significant details of the motor geometry and heat transfer characteristics are included. Though computationally efficient compared to FEA or CFD analysis as it requires solving a large set of differential algebraic equations, it is considered unsuitable for online deployment in the controller system [8]. In a reduced-order LPTN model, the heat transfer pathways are abstracted and thus can be represented with fewer nodes. They can be further classified as light grey (with 5–15 nodes) or dark grey (with 2–5 nodes) models and potentially are computationally lightweight. The representative thermal parameters in a grey-box model are not well known or possible to calculate using analytical equations [9,10]. For example, the heat transfer coefficients in the airgap and the end-cap regions

are nonlinearly dependent on the motor speed and the ambient temperature. Similarly, the heat transfer coefficient in the stator frame varies as per the ambient temperature. This makes the thermal parameters non-constant and linearly time-varying. The loss components such as winding loss, which are input to the LPTN model, can be calculated from the measured current and winding resistance data; however, the resistance values are temperature-dependent, which is a variable state in the thermal matrix. Furthermore, the core loss is not measurable. Therefore, the estimation of the core loss is made based on FEA methods or by subtracting the winding losses from the measured total losses. Hence, all errors in the determination of total and winding losses directly add up to an error in the iron loss values. In the above backdrop, empirically-based parametrization is a crucial step in these studies [9–15]. The parameter identification procedure can be stated as an optimization problem in which the values are varied until the used LPTN model gives the same results as the experimental ones.

Several gradient-based and non-gradient-based optimization methods have been explored in the past for identifying thermal parameters. Sciascera et al. [9] employed a calibration procedure based on a sequential quadratic programming iterative method for obtaining the uncertain thermal parameters in the LPTN. However, the computational cost of such tuning procedures is high due to the time-variant nature of the parameters. Hence, to improve computational efficiency, the dependence of the state matrix on the phase current is approximated with polynomial approximation. In the work presented by the authors [11], the parameter identification is conducted by using two determinist methods, the Gauss-Newton method and the Levenberg-Marquardt method, and one stochastic method, the Genetic Algorithm method. The results show good accuracy in predicting the temperature; however, the authors concluded that stochastic methods are more precise for a real-time model operation. Gedlu et al. [13] used particle swarm optimization (PSO) for the temperature estimation, minimizing the error against the empirical measurements. In addition to determining the thermal parameters, the coefficients for loss distributions are tuned in the optimization process to account for the loss-related uncertainties.

As described by Wallscheid [7], parameter identification can be performed in two ways: a local approach or a global approach. In the local approach, the scheduling variables that influence the thermal parameters are kept constant during the measurements. Huber et al. [12] presented a local approach for parametrizing a three-node LPTN model. A sequence of independent temperature measurement data is stored, varying the model inputs while keeping the scheduling variable constant. Each set of measurement data represents a linear time-invariant model cycle. The corresponding loss inputs are calculated with the available motor electrical control unit data such as motor speed and electric currents. The parameter identification approach was built on the idea of mapping the linear time-varying parameters to a set of time-invariant models operating within the boundary of the scheduling variable. Thus, a consistent parameter set for the whole operating region could be obtained with the adaptation of the relevant boundary conditions through various identification cycles. In contrast to the local approach, in the global approach, both the scheduling variables and the model inputs are simultaneously varied. Thus, the parameters are identified utilizing the comprehensive experimental data, leading to more robust results. Wallscheid [14] used a global approach to identifying the thermal parameters of a four-node LPTN model and the coefficients for the loss distribution. Extensive power-loss measurements were made in the complete operation range while keeping the temperature constant at the nodes. Additionally, the loss distribution among the nodes is interpreted as part of the parameter identification procedure for dealing with the uncertainty. A combination of PSO; a metaheuristic optimization method; and a gradient-based, sequential quadratic programming technique is used to deal with the nonlinear dependence of the parameters on the schedule variables.

While the global approach is more robust than the local approach in capturing all operating regions of the motor, it can also be problematic if the parameter landscape to be identified is large and highly nonlinear in nature [7]. Xiao and Griffo [15] in their work

present an online measurement-informed thermal parameter estimation using a recursive Kalman filter method. While a pulse width modulation-based estimation method is utilized for rotor temperature measurement, the temperatures in three nodes such as stator core, winding, and rotor are predicted. The input losses for the LPTN model are derived based on a model-based approach and with the use of FEA. The identification problem is formulated as a state observer with eight states and an Extended Kalman Filter is used to address the nonlinearity in the model. Three of the states correspond to the nodes' temperatures and the other five states represent the unknown thermal resistance parameters in the LPT network.

Though viable, the grey-box LPTNs require excessive motor measurement data to train a drastically reduced model order. However, the question remains whether such a high level of abstraction can accurately capture the nonlinearity and detail loss characteristics in a complex drive system application. Nonetheless, rapid progress in the applicability of machine learning (ML) techniques in varied domains and advancement in specialized embedded hardware platforms in the present scenario revolutionized the way temperature monitoring is performed. ML techniques are purely data-driven and employ neural networks to mimic the physical behavior of the involved system. A machine learning model can be trained offline or online empirically using data collected from test benches to parametrize a thermal model and estimate temperatures. These can exclude the expert knowledge of classic fundamental heat theory to do the task of estimating the temperature [16–18].

There are three approaches to machine learning methods: supervised, unsupervised, and reinforcement learning (RL). The applicability of the methods varies as per the applications and availability of the data. ML algorithms based on linear regression techniques have a low computational complexity. However, as linear regression is linear-time-invariant in nature, it does not capture the dynamics of a complex traction drive system. In the field of sequential learning tasks in highly dynamic environments, recurrent neural networks and convolutional neural networks are state-of-the-art in classification and estimation performance. Nogay [19] presented an artificial neural network (ANN) model to estimate the internal temperatures in the stator winding of a squirrel-cage induction motor supplied from a sinusoidal pulse width modulation inverter. The concept is to parameterize neural networks entirely on empirical data without being driven by the domain expertise. Wallscheid et al. in their work [20] explored a type of ANN technique, long short-term memories (LSTMs), to train for the real-time prediction of temperatures. The training procedure yielded a worst-case prediction error of 9–14 K which was found inferior compared to the LPTN approaches. However, the work will motivate future researchers to explore other ANN topology training methods to deal with the uncertainty of this problem. In the study conducted by Kirchgässner et al. [21], recurrent residual neural networks with memory blocks and convolutional neural networks with residual connections were empirically evaluated for predicting temperature profiles in the stator teeth, winding, and yoke. The model hyperparameter search was conducted sequentially via the Bayesian optimization method. The mean squared error and maximum absolute deviation performances of both these methods could match that of LPTN-model-based identification methods. While data-driven methods are proven to be effective in predicting the temperature, the tuned parameters are not interpretable and could not be designed with the low amount of model parameters as in an LPTN model at equal estimation accuracy. As a further development, to make expert knowledge-based calibration less desirable and to account for the uncertainties in the input power losses, Kirchgässner [22] proposed a deep-learning-based temperature model in which a thermal neural network is introduced, which unifies both consolidated knowledge in the form of heat-transfer-based LPTNs and a data-driven, non-linear function approximation with supervised machine learning.

Supervised ML models have also been considered for online electrical parameter estimation such as rotor resistance and mutual inductance in the control system of an induction motor [23,24]. In the presented work by the authors in [25], a simple two-layer ANN was trained by minimizing the error between the rotor flux linkages based on an IM analytical

voltage model and the output of the ANN-trained model. Feedforward and recurrent networks were used to develop an ANN as a memory for remembering the estimated parameters and for computing the electrical parameters during the transient state.

Machine learning applications are established as equally powerful tools in the controllers of motor drives [26–29]. In the work presented in [26], a hysteresis controller behavior was trained offline to an ANN topology to generate the desired switching pattern. Wishart et al. [27] worked on two ANN controllers trained to control the stator current and rotor speed adaptively to capture the induction machine dynamics. Implementations of ANN as speed controllers have been presented in [28,29]. The speed controllers were trained offline using simulation, but the weight and biases were updated online when the difference between the actual output and the targeted output of the ANN controller exceeds a preset value.

In the past, different neural network architectures have been explored for motor-drive control system applications. Recently, reinforcement learning has emerged as one of the most promising approaches for parameter estimation and as a substitute for controllers. RL techniques develop a control policy by learning through trial and error without the need for supervised data labeling. The algorithm relies on a reward function to guide the learning process. This allows for the continuous improvement of control policies based on feedback, showcasing the evolving landscape of ML applications in motor drive monitoring and control.

The reinforcement learning (RL)-based method is another promising data-driven technique explored in the field of the control of electric motor drives. An RL-enabled drive controller learns a control policy in a trial-and-error manner from the actual motor environment, thus avoiding the supervision of each data sample. The algorithm requires a reward function to receive the reward signals throughout the learning process. Thus, the control policy could be improved continuously based on the measurement feedback. Additionally, several other physical nonlinearities and parasitic influences from other drive system components can be adequately captured in the reinforced-based learning environment [30,31]. Although the learning process can be performed offline, the trained RL agent can connect to a control interface in the controller making it suitable for real-time implementation without demanding a drastic change in the embedded hardware [32,33]. Many exemplary works have been conducted that demonstrate its superiority in performance over a classic controller system. Bhattacharjee et al. [34] worked on an advanced deep reinforcement learning control with a dual Q learning approach for improving the performance of the current controller of a PMSM drive. The basis of the proposed controller is to learn the optimal action for the actor from the optimal voltage.

A twin-delayed deep deterministic-policy gradient (TD3) reinforcement-learning-based parametrization method is proposed by the authors in [35] to parametrize the thermal model for an induction traction motor. The RL framework includes the TD3 agent, a reward function, observations, and actions taken by the agent. Observations represent the data that the RL agent collected from the parameterized thermal model, which includes direct measurements such as motor speed (MS) and torque (MT). The RL agent evaluated the policies for the identification of optimized parameters. Indirect measurement data such as stator and rotor temperatures and model outputs were utilized for reward calculation. The reward function produced a value indicating how effective the action is and guided the RL agent toward achieving its goal.

This present work extends further to the applicability of multi-Agent reinforcement learning (MARL)-based framework comprising a set of RL agents such as soft actor-critic (SAC), and deep deterministic policy gradient (DDPG) besides the twin delayed deep deterministic policy gradient (TD3) for parameterizing the thermal model. Both measured and numerically estimated temperature data in several driving cycles are utilized to enhance the robustness of the parametrization and online adaptability.

The structure of this paper is outlined as follows: a background and related work is presented in Section 1. Section 2 introduces the multiple-agents reinforcement learning

framework developed for the optimization of the thermal model. Details regarding the dataset and the training methodology are also discussed in Section 2. Section 3 presents and analyzes the outcomes of the thermal model parameterization. Comparison of the proposed technique with the prior one is given in Section 4. Finally, Section 5 offers concluding thoughts and directions for future research.

2. Multi-Agent Reinforcement Learning Framework and Transfer Learning Approach

The multi-agent reinforcement learning (MARL)-based framework takes the idea of making smart choices from single-agent RL and can handle complex problems that get harder and harder as they grow. MARL makes smart decisions by trying different actions and seeing what works best. Each RL agent learns from its environment and the other agents to improve the whole team's results. This teamwork approach is efficient at solving complex problems that one agent alone could not handle, leading to smarter and more efficient solutions in real-world applications.

In synergizing the transfer learning approach, pre-trained reinforcement learning models are fine-tuned when applied to new similar driving cycles. In the proposed method, based on the source domain comprising several driving cycles on which the MARL models were initially trained, the model predicts the temperature in the target domain. The trained RL model includes important knowledge learned from the historical driving cycles' dataset. Thus, the model can utilize this existing or learned knowledge to make predictions on a new driving cycle efficiently. Nonetheless, the reinforcement learning pre-trained model's knowledge may not absolutely be aligned with the complexities of the new driving cycle data when the operation environment changes. Therefore, fine-tuning or change of a few parameters in the model must be performed to accurately predict the temperature. Thus, the pre-trained reinforcement learning models can be adjusted to optimize performance while taking advantage of the information acquired during pre-training.

Combining transfer learning with the MARL approach, the ability to handle the non-linearity of the thermal model and parasitic influences from other drive system components in the training process could be enhanced, thus making it suitable for real-time implementation. In addition, by transfer learning, the agents adapt and make informed decisions which potentially improves the overall performance and efficiency of the thermal model. This synergy between transfer learning and the MARL framework has the potential to develop an adaptive and robust solution that works for real-world dynamic scenarios.

2.1. The Reinforcement Learning Framework

A data-driven, reinforcement-learning-based parametrization method was proposed in our previous work [35] to estimate the thermal parameters of the thermal model of an induction traction motor as depicted in Figure 1. The thermal model has been parametrized as detailed in Appendix A. The RL agent uses a weighted combination of motor speed and torque as observation, Equation (2), and a weighted sum of the absolute differences between the measured and estimated temperatures as a reward, Equation (3), to produce the thermal parameters that maximize the reward value. The next sections explain various types of RL agents and the training process.

$$Observation = \{\omega_1 MS, \quad \omega_2 MT\} \tag{2}$$

$$Reward = \frac{-\omega_3}{\omega_4 |T_{sm} - T_{se}| + \omega_5 |T_{rm} - T_{re}| + \omega_6} \tag{3}$$

where MS, MT are motor speed and torque, respectively; $|T_{sm} - T_{se}|$ is the absolute value of the difference between measured stator temperature (T_{sm}) and estimated stator temperature (T_{se}); $|T_{rm} - T_{re}|$ is the absolute value of the difference between measured rotor temperature (T_{rm}) and estimated rotor temperature (T_{re}); and ω_i, $i = 1, 2, \ldots, 6$ are positive weights.

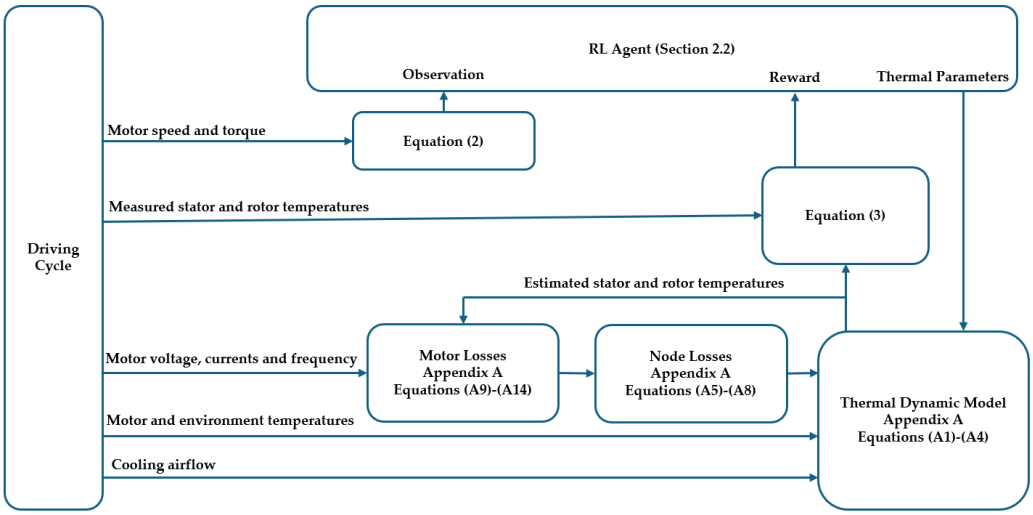


Figure 1. The reinforcement learning framework (adopted from [35], see Appendix A).

2.2. RL Multiple Agents Composition

The proposed data-driven approach employs three distinct reinforcement learning agents, twin delayed deep deterministic policy gradient (TD3), soft actor-critic (SAC), and deep deterministic policy gradient (DDPG). These agents are used to predict temperature based on the simulated data in nine operational driving cycles and the measurement data for one driving cycle.

2.2.1. Twin-Delayed Deep Deterministic Policy Gradient (TD3) Structure

The TD3 strategy employs a single-actor network with two critic networks as shown in Figure 2. These networks are deep neural networks that use observations, actions, and rewards which are gathered from the Experience Replay Buffer. The Experience Replay Buffer stores past experiences to allow the learning algorithm to benefit from a wide range of scenarios. The TD3 algorithm ensures that the actor and critic models are effectively trained to predict and evaluate the output of actions taken in the actual environment.

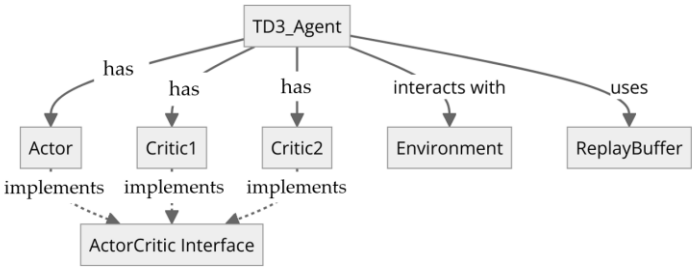


Figure 2. The architecture of the TD3 agent.

2.2.2. Deep Deterministic Policy Gradient (DDPG) Agent Structure

The DDPG agent combines the benefits of both policy gradient and value-based approaches. This framework excels in managing high-dimensional, continuous action spaces, making it particularly suitable for complex control tasks. The DDPG framework consists of four key components as shown in Figure 3 such as an actor network that proposes actions. A critic network evaluates the actions' potential reward and a target actor and critic network. These components use a replay buffer to store experiences and an update mechanism is used to combine target network parameters with the main networks.

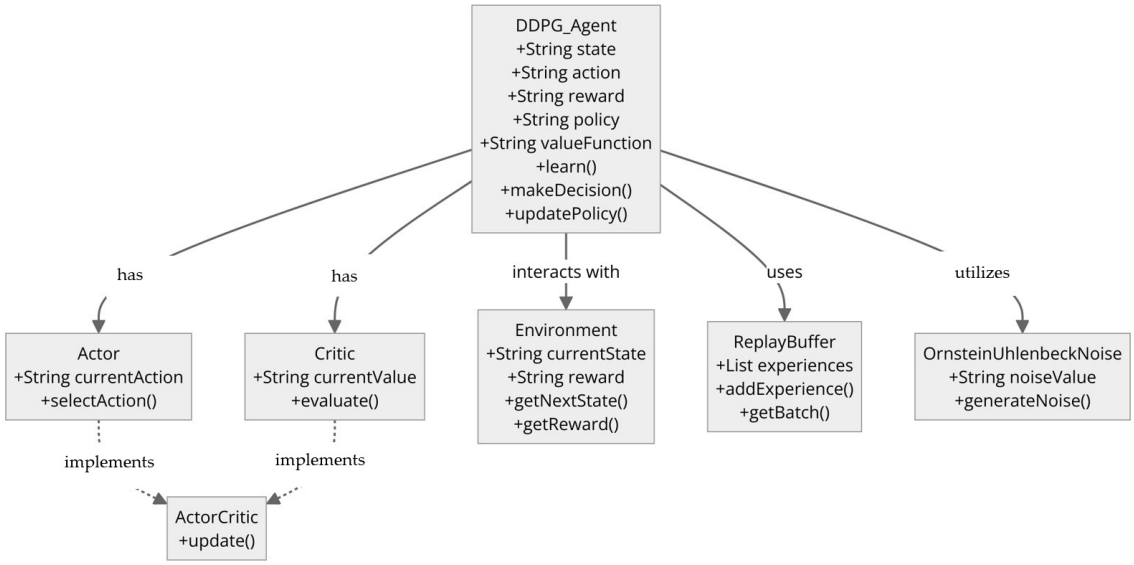


Figure 3. The architecture of the DDPG agent.

2.2.3. Soft Actor-Critic (SAC) Agent Structure

The SAC agent represents an advanced reinforcement learning strategy that emphasizes entropy in the policy for exploration. The SAC agent comprises a dual critic design to minimize the overestimation bias, as shown in Figure 4. The minimum value between two critics is considered to update the value network and actor. The use of entropy in the objective function boosts the agent to discover and exploit the optimal policy in complex environments.

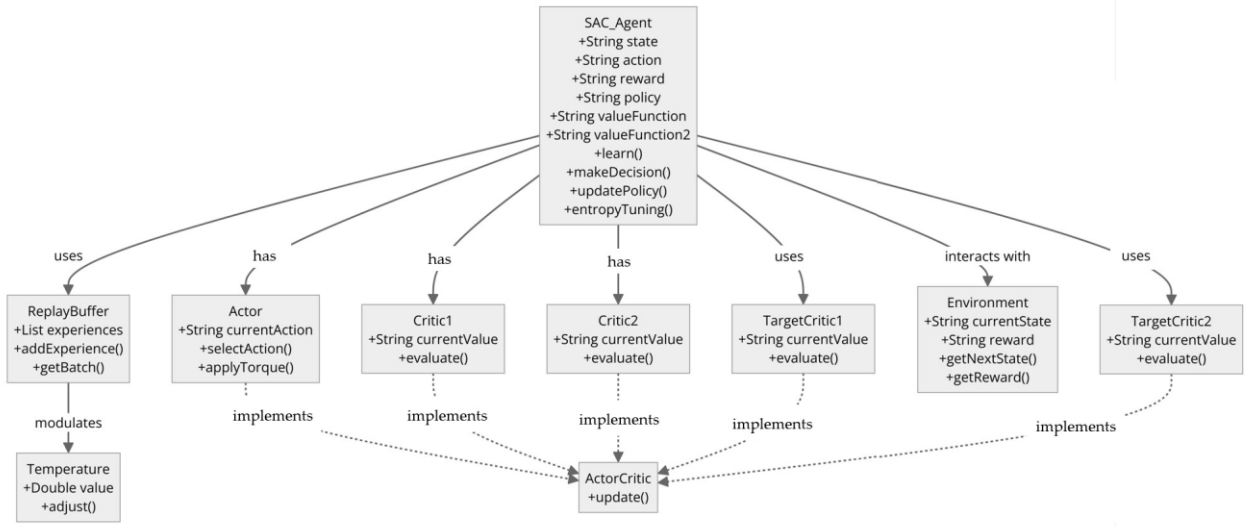


Figure 4. The architecture of the SAC agent.

The selection of the type of RL agent depends on certain criteria like data variation and the temporal aspects of each driving cycle data. For instance, the SAC agent uses an entropy-based exploration method for driving cycles with unpredictable temperature fluctuations. Conversely, TD3 and DDPG agents use deterministic policy gradients for more stable and predictable cycles. A TD3 algorithm is recognized for its efficacy in continuous environments that facilitate policy improvement and episodic training. The RL agents integrate the policy and learning strategy in order to map the observations to actions. The system uses both critic and actor networks in which the critic network predicts future rewards based on current actions and observations, whereas the actor selects the actions to

maximize these rewards. In the proposed approach, the architecture of the neural network, particularly the number of neurons, plays a critical role in the precision of stator and rotor temperature predictions.

2.3. RL Agent Training Process

2.3.1. Pre-Processing of Dataset

For the RL multiple-agent training, the dataset utilized was simulated data for the induction motor drive system under nine varied operational driving cycles and measured data in one drive cycle. The simulated dataset included motor speed, torque, airflow, stator current and frequency, motor voltage, and stator and rotor winding temperature, whereas the measured dataset included motor speed, torque, stator current, motor voltage, and stator temperatures. Due to the diversity in sampling frequencies and missing values within the dataset, a pre-processing step was carried out that involved the resampling and interpolation of the data to maintain the uniformity and completeness of the dataset. This dataset is employed within a parameterized model that simulates an unfamiliar environment for the agent.

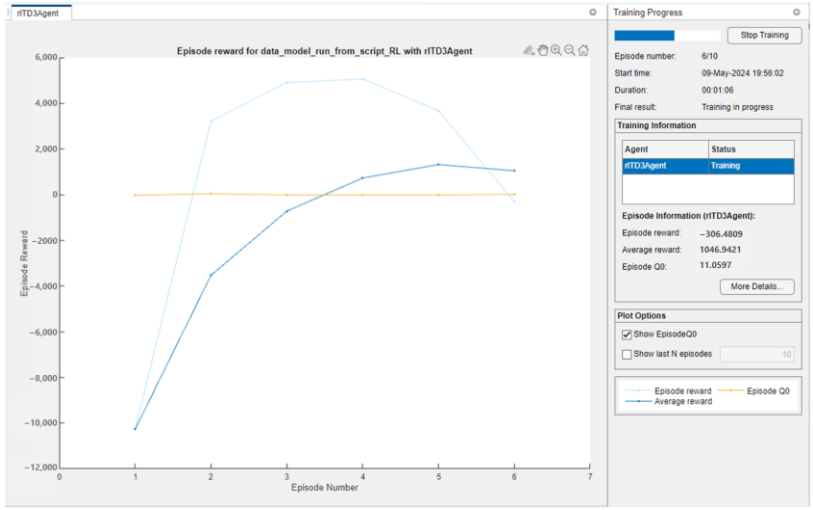
2.3.2. The Training Process

The training process of RL agents, as shown in Figure 5, is structured using an episodic approach, with each task considered as an episode. An episode includes the entire cycle of interactions between the agent and its environment. This episodic methodology allows the RL agent to systematically improve its decision-making capabilities by learning from diverse scenarios. Within this framework, the actor network determines an action based on the current observation and its anticipated Q-value. This selected action, the observation, and the reward received are stored in the Experience Replay Buffer. This buffer is then used to adjust the critic network parameters by reducing a loss function.

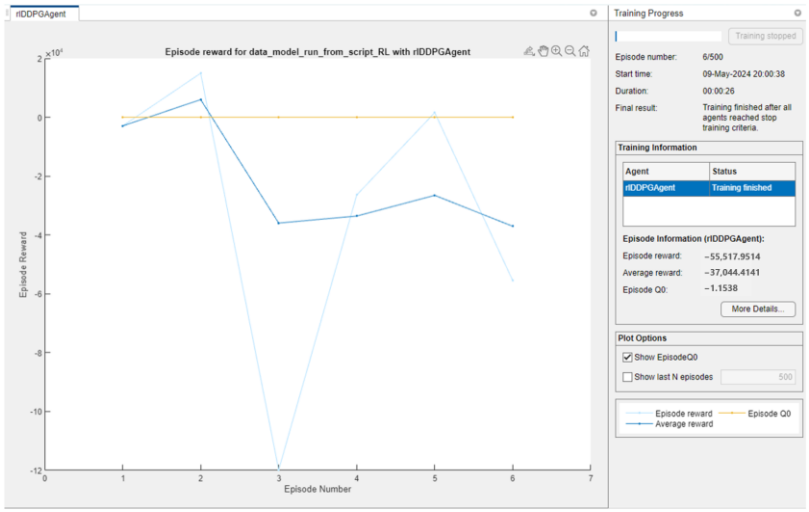
2.3.3. Transfer Learning from Pre-Trained RL Models

In addition to online training, the RL framework stored data from interactions within a simulated thermal model environment. After undergoing training on several controller driving cycles data, these pre-trained RL models and agents were then deployed to predict temperatures from new driving cycles data. The approach used statistical methods like mean, standard deviation, correlation, and frequency analysis to match new driving cycles with previously encountered ones. The analysis was performed on the basis of data from the same type of induction motor. When the new cycle data closely resembled the stored cycles data, the system selected an appropriate pre-trained model for temperature prediction, as shown in Figure 6. This saved computational time and resources by avoiding further training. However, if the new driving cycle appeared unique, the RL agent began training with this new controller-driving cycles data. This adaptive approach ensured that the model continuously evolved and learned from new motor usage patterns.

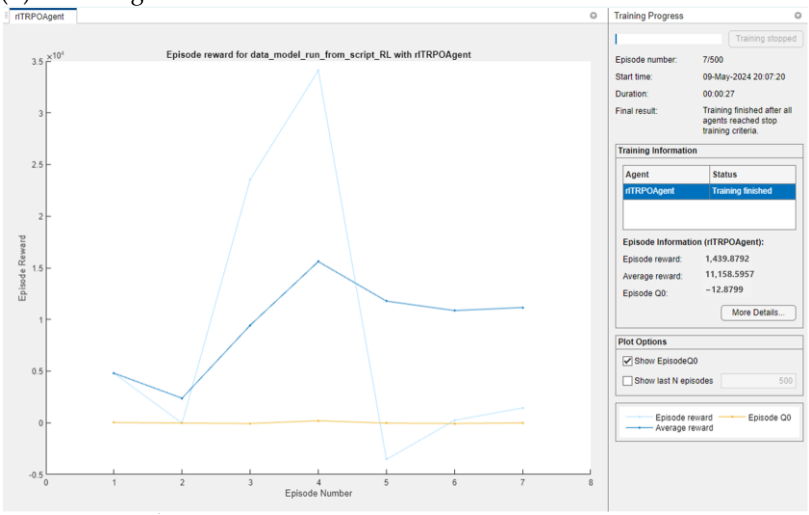
Experience replay is a critical technique utilized in the training of RL models, particularly in scenarios where acquiring new data is resource-intensive or new data arrives in batches, such as during driving cycles. By storing previously trained models along with their parameters, it can effectively "replay" these stored experiences when new data is received. This approach allowed the model to reinforce learning from past data without the need to undergo full retraining with each new batch of data. This method not only conserved computational resources but also enhanced the learning efficiency of the model. Experience replay capitalizes on fewer data samples by frequently revisiting and reprocessing stored data, thus amplifying the training process and minimizing the necessity for constant collection of vast quantities of new data. This strategy was instrumental in refining the model's performance over time, ensuring robustness and adaptability to new situations without the overhead of continuous training from scratch.



(**a**) TD3 Agent.



(**b**) DDPG Agent.



(**c**) SAC Agent.

Figure 5. The training process of: (**a**) TD3, (**b**) DDPG, and (**c**) SAC agents.

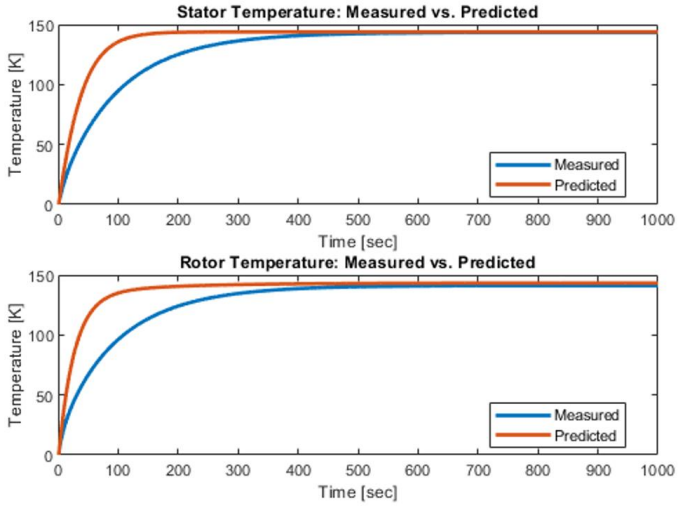


Figure 6. Performance of the selected pre-trained model for a new driving cycle.

3. Results & Discussion

3.1. Preprocessing the Dataset

The dataset represents the data recorded from the induction traction motor during the operation of nine different driving cycles. It is composed of all the signals generated from the driving cycle block shown in Figure 1, i.e., motor speed, torque, voltage, current, and frequency; stator, rotor and environment temperatures; and cooling airflow.

The data are recorded at different sampling frequencies with some missing values that require resampling the dataset and interpolating the missing values.

It should also be noted that the dataset is used to run the parameterized model, which is considered unknown to the RL agent, and to train the RL agent to produce the optimal thermal parameters.

3.2. Training the RL Agents

The training parameters that are crucial for the learning process of RL agents are outlined across multiple tables, each focusing on distinct driving cycles. This detailed breakdown of training parameters highlights the strategic and customized approach for training the RL agents.

Table 1 presents the parameters for the TD3 algorithm that is applied to driving cycles DC1, DC2, and DC3, specifying a limit of 10 episodes, with each episode consisting of up to 600 steps.

Table 1. Training parameters of the TD3 agent for driving cycles DC1, DC2, and DC3.

Property	Value
max episodes	10
max steps per episodes	600
average window length	500
stop training value	−10
agent sample time	0.1 s

Table 2 describes the training parameters relevant to the DDPG algorithm used for driving cycles DC4, DC6, DC7, and DC9. For the DC9, some different training parameters were used, such as the maximum steps per episode, averaging window lengths, and stop values.

Table 2. Training parameters of the DDPG agent for driving cycles DC4, DC6, DC7, and DC9.

Property	Value (DC4, DC6, DC7)	Value (DC9)
max episodes	1000	1000
max steps per episodes	600	20
average window length	500	20
stop training value	−10	−740
agent sample time	0.1 s	0.1 s

Table 3 specifies the training parameters for the DDPG algorithm for DC 9, closely following the stator's configuration in DC 8. This includes 1000 maximum episodes, 20 steps per episode, an averaging window of 50, a stop value of −740, and a sample time of 0.1 s.

Table 3. Training parameters of the DDPG agent for driving cycles DC8.

Property	Value (Rotor)	Value (Stator)
max episodes	1000	1000
max steps per episodes	600	20
average window length	500	50
stop training value	−10	−740
agent sample time	0.1 s	0.1 s

3.3. Validating the Trained Agents

The validation of the RL agent involved regularly evaluating the policy it implemented in the actual environment. The results of this process, in which it compared the observed and predicted temperatures for tool and measured driving cycles, are shown in the following figures.

Figure 7 presents the outcomes from deploying a TD3 reinforcement learning agent for temperature predictions across three specific driving cycles. This figure shows the temperature predictions over time for the stator and the rotor of an induction motor from tool data. In Figure 7, a blue line represents the actual measured temperature data, and an orange line shows the predicted temperature by the TD3 agent. The close match between measured data and the predictions made by the agents in three DC data indicate that the TD3 agent successfully learned and accurately estimated the thermal dynamics of induction motors. Furthermore, the precision of the temperature prediction demonstrates the agent's capability to generalize well from the training data.

Figure 7 also illustrates the performance of a predictive model for DC5 stator and rotor temperatures using the SAC RL agent. TD3 and DDPG agents were employed to get the optimal thermal parameters and temperature predictions from DC5. However, the SAC agent only provided improved temperature predictions for the stator, while it struggles to achieve accurate predictions for the rotor temperature. Figure 7 highlights the fluctuations and spikes in model predictions for stator temperature and indicates the less predictive accuracy of the model in the case of rotor temperature prediction. Optimization is required for DC5 for the thermal management and operational efficiency of electric motors. Rotor temperature discrepancies highlighted in the figures underscore the importance of analyzing thermal management strategies. The rotor, being the rotating part, may experience different levels of heat generation compared to the stator, primarily due to friction, air resistance, and the electrical losses that occur within the rotating component. Furthermore, cooling mechanisms, such as airflow have varying effectiveness on the rotor and stator due to their accessibility. The material properties of the rotor and stator, including thermal conductivity and specific heat capacity, also play significant roles in how heat is generated, absorbed, and transferred. Additionally, operational conditions, such as speed, can further influence the temperature disparity.

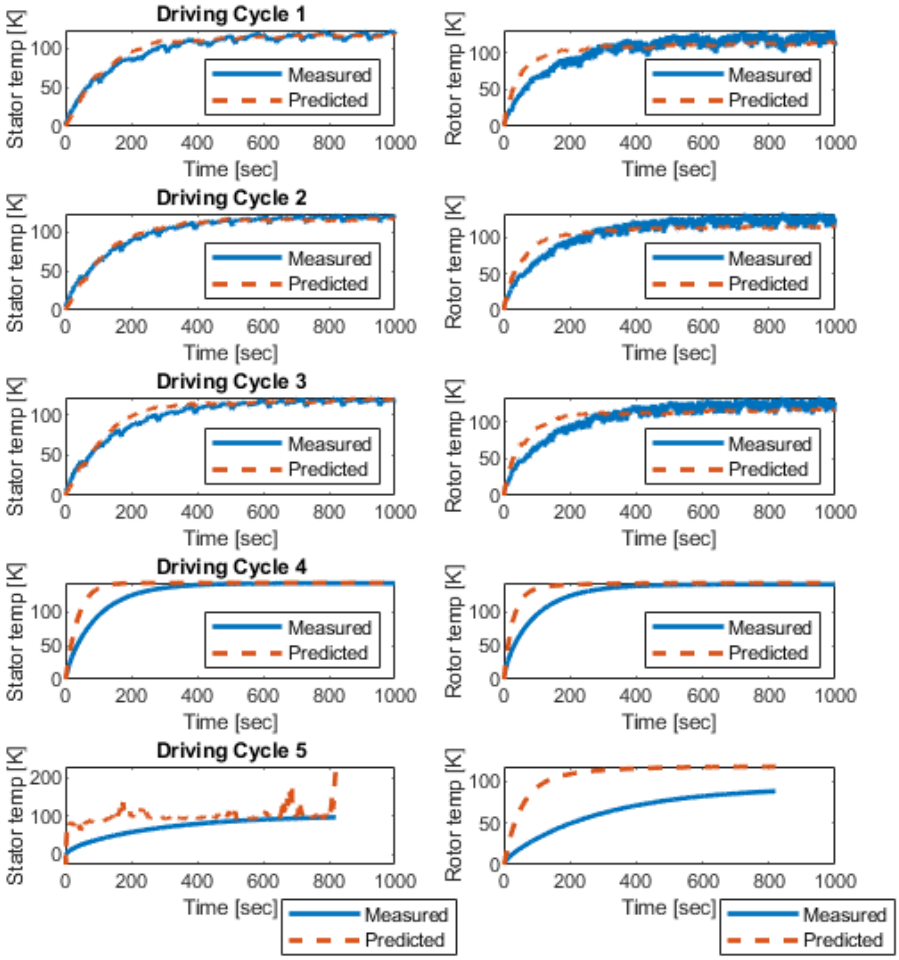


Figure 7. Stator and rotor temperature predictions of driving cycles (DC): DC1, DC2, DC3 using the TD3 agent, DC4 using DDPG, and DC5 using SAC.

Figure 8 displays the stator and rotor temperature predictions of the DDPG reinforcement learning agent on four distinct driving cycles data labeled as cycles DC6, DC7, DC8, and DC9. Figure 8 depicts the DDPG agent's prediction capabilities in aligning its temperature estimation closely with the actual data over time. Additionally, detailed insights highlight that the rotor temperature prediction of DC6 and stator temperature prediction of DC7 were not accurate as still there was room for improvement, and fine-tuning of the deep neural network is required to get approximate predictions from both driving cycles data.

Figure 9 depicts the predicted stator winding temperatures from measured data. The SAC agent was employed for stator winding temperature prediction with and without the inclusion of airflow information. The prediction of stator temperature in the first figure, which includes airflow, displays more precise predictions of the operating conditions, whereas the 2nd figure represents the predicted temperatures without considering the airflow information. The absence of airflow data in the thermal model results in less accurate predictions, in which the agent is unable to capture the immediate effects of varying stator temperature. Adding airflow information to the proposed thermal models matters because it affects the prediction of stator winding temperature.

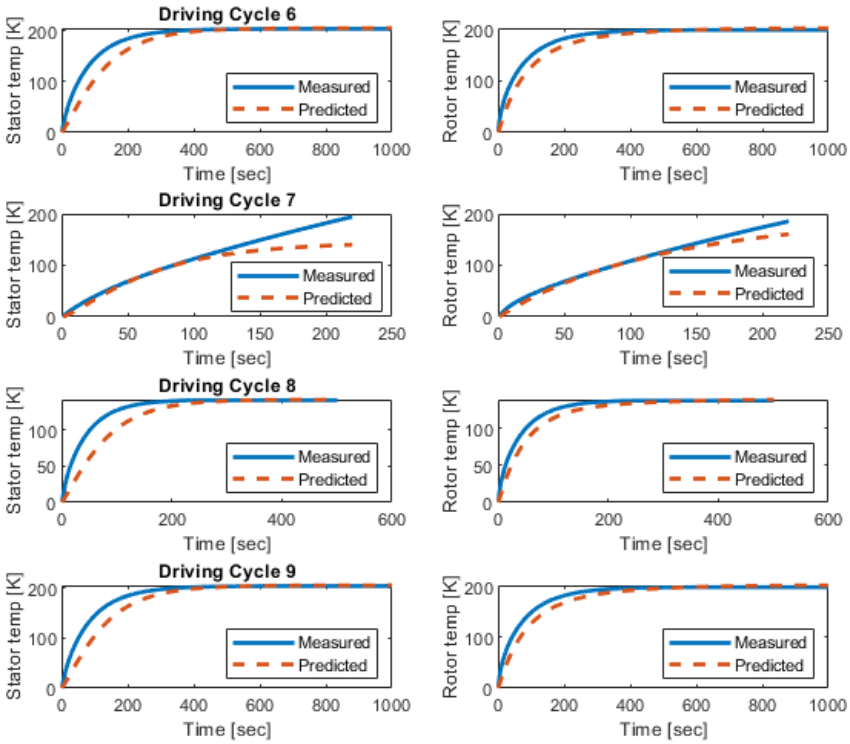
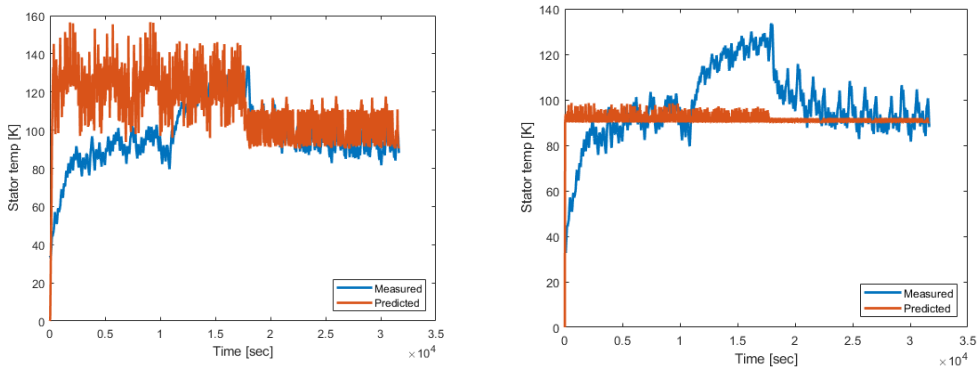


Figure 8. Stator and rotor temperature predictions of driving cycles (DC): DC6, DC7,DC8, and DC9 using the DDPG agent.



(**a**) (**b**)

Figure 9. Stator and rotor temperature predictions and results of the driving cycle of measured data using the SAC agent: (**a**) with airflow information and (**b**) without airflow information.

4. Contrasting the Proposed Technique with the Prior One

The proposed technique is compared with [35] as both techniques focus on accurate temperature estimation for railway propulsion systems to ensure optimal operation. In the proposed method, a data-driven, reinforcement-learning-based parametrization approach is proposed to estimate the thermal model parameters. This approach involves training RL agents using data from various driving cycles. Different strategies were developed to manage the thermal behaviors of controller driving cycles effectively under different operational conditions. The capability of RL agents was emphasized to address driving variability and to accurately reflect the motor's thermal behavior. It also presented an offline mode, in which pre-trained models were utilized to predict temperatures from driving cycle data. On the other hand, the approach proposed in [35] uses a physics-based thermal model of the induction traction motor. This approach demonstrates the competence of RL agents in developing strategies for managing thermal behaviors under different operational conditions and accurate temperature estimation in induction traction motors. For validating

the thermal model, our proposed method employs data from nine driving cycles as well as measured data to ensure that the model's accuracy encompasses various scenarios. In contrast, [35] validated their model with only two driving cycles. Although this may appear to be a smaller dataset, it is worth noting that [35] did not develop an offline model, which is crucial for computational efficiency, especially when utilizing pre-trained models for precise temperature predictions.

5. Conclusions

This paper presents a physics-based and data-driven multi-agent reinforcement learning approach to predict the temperature of an induction traction motor from recorded driving cycles. A physics-based thermal model of the induction traction motor was used with the simulated data to generate all the required signals that were needed to train RL agents across various driving cycles The trained reinforcement learning agents demonstrated good proficiency in devising strategies for managing thermal behaviors under different operational conditions. In the offline mode, pre-trained models were utilized to predict the temperature from several driving cycles data.

To handle nonlinearities and parasitic influences from other drive system components and make the multi-agent reinforcement learning approach suitable for real-time implementation, transfer learning was integrated into the approach, in which pre-trained reinforcement learning models are fine-tuned for new driving cycles. By enabling agents to adapt and make informed decisions through transfer learning, the overall performance and efficiency of the thermal model were significantly improved, resulting in an adaptive and robust solution applicable to real-world dynamic scenarios.

Transfer learning allows us to use domain knowledge gained from pre-trained models and historical data. The reinforcement learning agents can be effectively fine-tuned and transferred to this knowledge in order to optimize the temperature predictions for new driving cycles. The integration of the transfer learning technique in the proposed approach also demonstrates increased adaptability, which is crucial for accurate temperature prediction across diverse driving cycle scenarios. Synergizing transfer learning in the proposed approach goes beyond traditional approaches and changes thermal management strategies in railway propulsion systems. As they are confined to a controlled lab environment, this approach focuses more on adaptability and efficient thermal management methods.

The integration of statistical techniques and clustering to identify relevant driving cycles for offline prediction further emphasized the comprehensive nature of the approach. However, there are a few limitations of the proposed approach that must be addressed for the training of RL multi-agents to be computationally intensive and will require further investigation to generalize across different motor types and conditions.

Author Contributions: Conceptualization and methodology, F.M., A.F. and S.S.; software, F.M. and A.F.; validation, F.M.; data curation, S.S.; writing—original draft preparation, F.M.; writing—review and editing, A.F. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge the support for this research provided by the AIDOaRt project, which is financed by the ECSEL Joint Undertaking (JU) under grant agreement No 101007350.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality agreement.

Conflicts of Interest: Author Smruti Sahoo was employed by the company Alstom. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Parametrizing the Thermal Model

From a thermal point of view, the motor is modeled with four nodes: stator winding (node 1), stator core (node 2), rotor winding (node 3), and rotor core (node 4). The thermal equivalent network is illustrated in Figure A1 with thermal capacitances; it is connected to a power source and has thermal conductance among the nodes and to the cooling air.

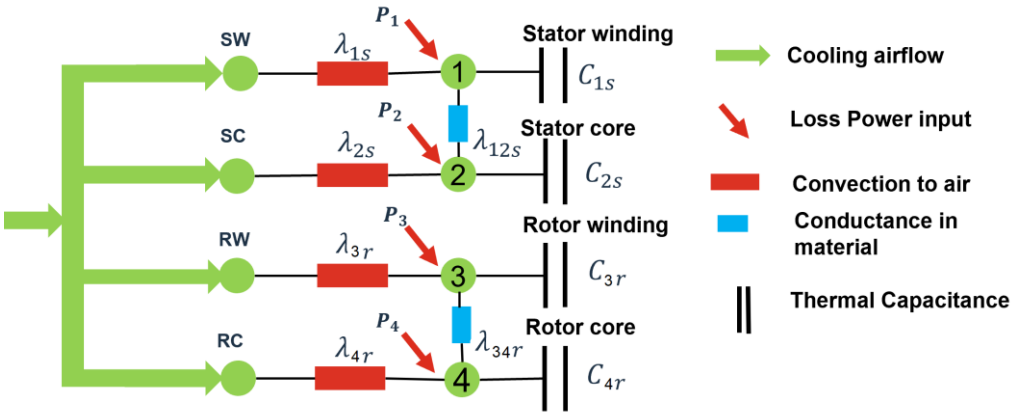


Figure A1. Lumped-parameter thermal network model.

Values of the thermal capacitance C_{1s}, C_{2s}, C_{3r}, and C_{4r} are calculated analytically from the geometric and material information of the motor, where the indices 1s, 2s, 3r, and 4r refer to Node 1-Stator, Node 2-Stator, Node 3-Rotor, and Node 4-Rotor respectively. The capacitance for stator yoke C_{1s} is the sum of the capacitance of stator housing, stator back iron, stator tooth, and flange mounting [36]. The stator winding capacitance C_{2s} includes the capacitance for the stator winding and the end winding capacitances. The capacitance for stator yoke C_{1r} is the sum of the capacitance of the rotor yoke and rotor bars. The rotor winding capacitance C_{2r} includes the capacitance for the rotor winding and the end winding capacitances. The thermal conductance λ_{1s}, λ_{2s}, λ_{3r}, and λ_{4r} vary with the airflow due to the convection. The model shown in Figure A1 can be represented mathematically by the following first-order differential system:

$$P_1 = C_{1s}\frac{dT_1}{dt} + \lambda_{1s}(T_1 - T_{env}) + \lambda_{12s}(T_1 - T_2), \quad \text{(A1)}$$

$$P_2 = C_{2s}\frac{dT_2}{dt} + \lambda_{2s}(T_2 - T_{env}) + \lambda_{12s}(T_2 - T_1), \quad \text{(A2)}$$

$$P_3 = C_{3r}\frac{dT_3}{dt} + \lambda_{3r}(T_3 - T_{env}) + \lambda_{34r}(T_3 - T_4), \quad \text{(A3)}$$

$$P_4 = C_{4r}\frac{dT_4}{dt} + \lambda_{4r}(T_4 - T_{env}) + \lambda_{34r}(T_4 - T_3), \quad \text{(A4)}$$

where T_i is the temperature at the corresponding node i. The temperatures of the cooling air at the four nodes (marked as SW, SC, RW, and RC in Figure A1) are assigned to the environmental (or ambient) temperature T_{env}.

The losses at the four nodes in Figure A1 are distributed as shown in Table A1 and they can be calculated as follows:

$$P_1 = K_{stemp}P_{cu1} + K_{stray}P_{stray} + K_{harm}P_{harm}, \quad \text{(A5)}$$

$$P_2 = K_{Pfe}P_{fe}, \quad \text{(A6)}$$

$$P_3 = K_{rtemp}P_{cu2} + (1 - K_{stray})P_{stray} + (1 - K_{harm})P_{harm}, \quad \text{(A7)}$$

$$P_4 = \left(1 - K_{Pfe}\right)P_{fe}, \quad \text{(A8)}$$

where P_{cu1}, P_{cu2}, P_{stray}, P_{harm}, and P_{fe} are the stator copper loss, rotor copper loss, stray loss, harmonic loss, and iron loss, respectively. The coefficients K_{stemp}, K_{rtemp}, K_{stray}, K_{harm} and K_{Pfe} are the corresponding loss coefficients.

Table A1. Distribution of losses in the LPTN model (x denotes a connection between the corresponding row and column).

Node	Winding Losses	Stray Losses	Harmonic Losses	Iron Losses
1	x	x	x	
2				x
3	x	x	x	
4				x

The losses in Equations (A5)–(A8) can be calculated as follows [8,37–39]:

$$P_{cu1} = R_1 \, I_1^2, \tag{A9}$$

$$P_{cu2} = R_{21} \, I_{21}^2, \tag{A10}$$

$$P_{stray} = P_{SUP} \left(\frac{f}{f_{nom}} \right)^{1.5} \left(\frac{I_1}{I_{1,nom}} \right)^2, \tag{A11}$$

$$P_{fe} = K_f \, f^{\alpha} B_{max}^{\beta}, \tag{A12}$$

where I_1, and I_{21} are the stator and rotor currents, respectively, and R_1 and R_2 are the stator and rotor winding resistances, respectively, which depend on the temperature according to following equations:

$$R_1 = R_{1,20} * (1 + \alpha_{R1} * (T_1 - 20)), \tag{A13}$$

$$R_{21} = R_{21,20} * (1 + \alpha_{R2} * (T_3 - 20)), \tag{A14}$$

where $R_{1,20}$ and $R_{21,20}$ are the stator and rotor winding resistances at 20 °C, and α_{R1} and α_{R21} are the temperature coefficients of the stator and rotor, respectively. In Equation (A11), f is the stator frequency with a nominal value f_{nom}, I_1 is the stator current with a nominal value $I_{1,nom}$, and P_{SUP} is the equivalent-rated input power. In Equation (A12), K_f is a constant that depends on the material properties and the core geometry, f is the frequency of the magnetic field, B_{max} is the peak magnetic flux density in the core, and α and β are empirically determined constants. The harmonic losses P_{harmo} are measured at a few operation points and included as a look-up table in the loss model.

References

1. Fathy Abouzeid, A.; Guerrero, J.M.; Endemaño, A.; Muniategui, I.; Ortega, D.; Larrazabal, I.; Briz, F. Control strategies for induction motors in railway traction applications. *Energies* **2020**, *13*, 700. [CrossRef]
2. Hannan, M.A.; Ali, J.A.; Mohamed, A.; Hussain, A. Optimization techniques to enhance the performance of induction motor drives: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1611–1626. [CrossRef]
3. Lemmens, J.; Vanassche, P.; Driesen, J. Optimal control of traction motor drives under electrothermal constraints. *IEEE J. Emerg. Sel. Top. Power Electron.* **2014**, *2*, 249–263. [CrossRef]
4. Dorrell, D.G. Combined thermal and electromagnetic analysis of permanent-magnet and induction machines to aid calculation. *IEEE Trans. Ind. Electron.* **2008**, *55*, 3566–3574. [CrossRef]
5. Ramakrishnan, R.; Islam, R.; Islam, M.; Sebastian, T. Real time estimation of parameters for controlling and monitoring permanent magnet synchronous motors. In Proceedings of the 2009 IEEE International Electric Machines and Drives Conference, Miami, FL, USA, 3–6 May 2009; pp. 1194–1199.
6. Wilson, S.D.; Stewart, P.; Taylor, B.P. Methods of resistance estimation in permanent magnet synchronous motors for real-time thermal management. *IEEE Trans. Energy Convers.* **2010**, *25*, 698–707. [CrossRef]
7. Wallscheid, O. Thermal monitoring of electric motors: State-of-the-art review and future challenges. *IEEE Open J. Ind. Appl.* **2021**, *2*, 204–223. [CrossRef]
8. Kral, C.; Haumer, A.; Lee, S.B. A practical thermal model for the estimation of permanent magnet and stator winding temperatures. *IEEE Trans. Power Electron.* **2013**, *29*, 455–464. [CrossRef]

9. Sciascera, C.; Giangrande, P.; Papini, L.; Gerada, C.; Galea, M. Analytical thermal model for fast stator winding temperature prediction. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6116–6126. [CrossRef]
10. Zhu, Y.; Xiao, M.; Lu, K.; Wu, Z.; Tao, B. A simplified thermal model and online temperature estimation method of permanent magnet synchronous motors. *Appl. Sci.* **2019**, *9*, 3158. [CrossRef]
11. Guemo, G.G.; Chantrenne, P.; Jac, J. Parameter identification of a lumped parameter thermal model for a permanent magnet synchronous machine. In Proceedings of the 2013 International Electric Machines & Drives Conference, Chicago, IL, USA, 12–15 May 2013; pp. 1316–1320.
12. Huber, T.; Peters, W.; Böcker, J. Monitoring critical temperatures in permanent magnet synchronous motors using low-order thermal models. In Proceedings of the 2014 International Power Electronics Conference (IPEC-Hiroshima 2014-ECCE ASIA), Hiroshima, Japan, 18–21 May 2014; IEEE: Piscataway, NJ, USA, 2014.
13. Gedlu, E.G.; Wallscheid, O.; Böcker, J. Permanent magnet synchronous machine temperature estimation using low-order lumped-parameter thermal network with extended iron loss model. In Proceedings of the 10th International Conference on Power Electronics, Machines and Drives (PEMD 2020), Online Conference, 15–17 December 2020; IET: London, UK, 2020.
14. Wallscheid, O.; Böcker, J. Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors. *IEEE Trans. Energy Convers.* **2015**, *31*, 354–365. [CrossRef]
15. Xiao, S.; Griffo, A. Online thermal parameter identification for permanent magnet synchronous machines. *IET Electr. Power Appl.* **2020**, *14*, 2340–2347. [CrossRef]
16. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark. *IEEE Trans. Energy Convers.* **2021**, *36*, 2059–2067. [CrossRef]
17. Zhang, S.; Wallscheid, O.; Porrmann, M. Machine learning for the control and monitoring of electric machine drives: Advances and trends. *IEEE Open J. Ind. Appl.* **2023**, *4*, 188–214. [CrossRef]
18. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 318–323.
19. Nogay, H.S. Prediction of internal temperature in stator winding of three-phase induction motors with ann. *Eur. Trans. Electr. Power* **2011**, *21*, 120–128. [CrossRef]
20. Wallscheid, O.; Kirchgässner, W.; Böcker, J. Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1940–1947.
21. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors. In Proceedings of the 2019 IEEE International Electric Machines & Drives Conference (IEMDC), San Diego, CA, USA, 12–15 May 2019; IEEE: Piscataway, NJ, USA, 2019.
22. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105537. [CrossRef]
23. Wlas, M.; Krzeminski, Z.; Toliyat, H.A. Neural-network-based parameter estimations of induction motors. *IEEE Trans. Ind. Electron.* **2008**, *55*, 1783–1794. [CrossRef]
24. Rafaq, M.S.; Jung, J.W. A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4747–4758. [CrossRef]
25. Harashima, F.; Demizu, Y.; Kondo, S.; Hashimoto, H. Application of neutral networks to power converter control. In Proceedings of the Conference Record of the IEEE Industry Applications Society Annual Meeting, San Diego, CA, USA, 1–5 October 1989; IEEE: Piscataway, NJ, USA, 1989; pp. 1086–1091.
26. Wishart, M.T.; Harley, R.G. Identification and control of induction machines using artificial neural networks. *IEEE Trans. Ind. Appl.* **1995**, *31*, 612–619. [CrossRef]
27. Rahman, M.A.; Hoque, M.A. On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors. *IEEE Trans. Energy Convers.* **1998**, *13*, 311–318.
28. Yi, Y.; Vilathgamuwa, D.M.; Rahman, M.A. Implementation of an artificial-neural-network-based real-time adaptive controller for an interior permanent-magnet motor drive. *IEEE Trans. Ind. Appl.* **2003**, *39*, 96–104.
29. Zhang, Q.; Zeng, W.; Lin, Q.; Chng, C.B.; Chui, C.K.; Lee, P.S. Deep reinforcement learning towards real-world dynamic thermal management of data centers. *Appl. Energy* **2023**, *333*, 120561. [CrossRef]
30. Egan, D.; Zhu, Q.; Prucka, R. A Review of Reinforcement Learning-Based Powertrain Controllers: Effects of Agent Selection for Mixed-Continuity Control and Reward Formulation. *Energies* **2023**, *16*, 3450. [CrossRef]
31. Zhang, Y.; Huang, J.; He, L.; Zhao, D.; Zhao, Y. Reinforcement learning-based control for the thermal management of the battery and occupant compartments of electric vehicles. *Sustain. Energy Fuels* **2024**, *8*, 588–603. [CrossRef]
32. Book, G.; Traue, A.; Balakrishna, P.; Brosch, A.; Schenke, M.; Hanke, S.; Kirchgässner, W.; Wallscheid, O. Transferring online reinforcement learning for electric motor control from simulation to real-world experiments. *IEEE Open J. Power Electron.* **2021**, *2*, 187–201. [CrossRef]
33. Traue, A.; Book, G.; Kirchgässner, W.; Wallscheid, O. Toward a reinforcement learning environment toolbox for intelligent electric motor control. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 919–928. [CrossRef]

34. Bhattacharjee, S.; Halder, S.; Balamurali, A.; Towhidi, M.; Iyer, L.V.; Kar, N.C. An advanced policy gradient based vector control of PMSM for EV application. In Proceedings of the 2020 10th International Electric Drives Production Conference (EDPC), Ludwigsburg, Germany, 8–9 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
35. Fattouh, A.; Sahoo, S. Data-driven reinforcement learning-based parametrization of a thermal model in induction traction motors. In Proceedings of the 64th International Conference of Scandinavian Simulation Society, SIMS 2023, Västerås, Sweden, 25–28 September 2023; pp. 310–317.
36. Wöckinger, D.; Bramerdorfer, G.; Drexler, S.; Vaschetto, S.; Cavagnino, A.; Tenconi, A.; Amrhein, W.; Jeske, F. Measurement-based optimization of thermal networks for temperature monitoring of outer rotor PM machines. In Proceedings of the 2020 IEEE Energy Conversion Congress and Exposition (ECCE), Detroit, MI, USA, 11–15 October 2020; pp. 4261–4268.
37. Filizadeh, S. *Electric Machines and Drives: Principles, Control, Modeling, and Simulation*; CRC Press: Boca Raton, FL, USA, 2013.
38. Maroteaux, A. *Study of Analytical Models for Harmonic Losses Calculations in Traction Induction Motors*; KTH, School of Electrical Engineering (EES): Stockholm, Sweden, 2016.
39. Roshandel, E.; Mahmoudi, A.; Kahourzade, S.; Yazdani, A.; Shafiullah, G. Losses in efficiency maps of electric vehicles: An overview. *Energies* **2021**, *14*, 7805. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.