Estimating Temperature in a Permanent-Magnet Synchronous Motor Using Hammerstein and Nonlinear Autoregressive Models Initialized via Thermal Networks

Erick Axel Martinez-Ríos, Irving S. Aguilar-Zamorate, Saulius Pakštys, Renato Galluzzi, *Senior Member, IEEE*, and Nicola Amati

***Abstract*—Monitoring the temperature of permanent-magnet synchronous motors is crucial to prevent failures in sensitive components such as windings and permanent magnets. In this respect, machine learning techniques have been used to generate models to estimate the temperature of rotor and stator hotspots. However, the effective use of data-driven methods requires large datasets, extensive training time, and substantial computational power. Moreover, machine learning methods mostly operate with a black-box approach; they do not account for the physics of the system to be modeled. This paper proposes and compares Hammerstein and nonlinear autoregressive exogenous models to estimate the temperature of the permanent magnets and windings of an out-runner permanent-magnet synchronous motor. A linear time-invariant component, used for both the Hammerstein and nonlinear autoregressive exogenous models, is initialized via a previously identified fourth-order lumped parameter thermal network. This model accounts for the thermal behavior of the machine. The nonlinear component is modeled via a neuron sigmoid network. Results show that the Hammerstein model achieves a lower mean squared error for the winding temperature estimation than the nonlinear autoregressive exogenous model. The opposite is true for the magnet temperature estimation.**

***Index Terms*—temperature estimation, permanent-magnet synchronous motors, Hammerstein model, nonlinear autoregressive exogenous models, lumped parameter thermal network**

I. Introduction

PERMANENT-magnet synchronous motors (PMSMs) have become widely used in various industrial applications, including wind power generation, electrical aircraft propulsion systems, robotics, and electric vehicle powertrains, due to their high torque and power density, among other features [1]. Despite their advantages, PMSMs are not exempt from electrical, mechanical, or magnetic faults [2]. In particular, magnetic faults are related to the demagnetization of the rotor permanent magnets (PMs), which usually occurs at high temperature. This phenomenon implies a reduction in the PM remanence, impacting the machine's torque capability [3], [4]. Furthermore, short circuits could affect windings due to melting insulation caused by thermal stress [5], [6]. In light of these thermal limitations, applications of PMSMs with dynamically changing operating conditions, such as automation or vehicle traction, need adequate monitoring of thermal response to prevent thermal overloading [7].

The above has motivated the development of various methods for estimating the temperature of PMSMs, including contact and non-contact sensors, finite element analysis (FEA) and computational fluid dynamics (CFD), electrical models, thermal models, and data-driven models [7]. Sensors introduce the added complexity of structural modifications, and in turn, high costs. Additionally, temperature measurement in a rotating frame is a complex task [8]. CFD and FEA are computationally expensive and time-consuming, which makes them unsuitable for real-time monitoring [9], [10]. Electrical models can be used to track the temperature inside the PMSM by estimating temperature-sensitive parameters (e.g., electrical resistivity of copper or remanent flux density of PMs). The major limitation of such methods is the effect of minor inaccuracies in estimating electrical parameters, leading to significant temperature estimation errors [11]. PM materials like NdFeB have low-temperature sensitivity, and the ohmic voltage drop in high-power machines is negligible when compared to the terminal voltage [12], [13]. Thermal models, such as low-order lumped parameter thermal networks (LPTNs), demand collecting estimation data in a sufficient range of operational conditions, as they consist of varying parameters that depend on the PMSM speed, geometry, and cooling conditions [14].

Data-driven techniques have become a popular approach for estimating PMSM temperature. Unlike thermal or electrical models, they do not require prior knowledge of motor characteristics to generate a model that describes thermal behavior. These methods can be categorized into static and dynamic approaches. Static methods do not rely on previous input or output values and include regression trees, multilayer perceptrons, ordinary least squares, support vector regression, and

Manuscript received # Month 2025; revised # Month #

Erick Axel Martinez-Ríos was with the School of Engineering and Sciences, Tecnologico de Monterrey, Mexico City 14380, Mexico e-mail: a01331212@tec.mx

Irving S. Aguilar-Zamorate was with the School of Engineering and Sciences, Tecnologico de Monterrey, Mexico City 14380, Mexico e-mail: a01336855@tec.mx

Saulius Pakštys was with the Center for Automotive Research and Sustainable Mobility, Politecnico di Torino, Turin 10129, Italy e-mail: saulius.pakstys@polito.it

Renato Galluzzi was with the School of Engineering and Sciences, Tecnologico de Monterrey, Mexico City 14380, Mexico e-mail: renato.galluzzi@tec.mx

Nicola Amati was with the Center for Automotive Research and Sustainable Mobility, Politecnico di Torino, Turin 10129, Italy e-mail: nicola.amati@polito.it

k-nearest neighbor, which have been applied to PMSM temperature estimation [15], [16]. In contrast, dynamic methods consider the dynamic behavior of the system by incorporating previous input and output values. These include recurrent neural networks, long short-term memory networks, temporal convolutional neural networks, nonlinear autoregressive exogenous networks (NARX), and thermal neural networks, all of which have been used for PMSM temperature estimation [17]–[20].

Despite the prominent use of data-driven techniques to model the thermal behavior of PMSMs, these methods have notable disadvantages. First, gathering training data of acceptable quality for fitting is time-consuming and unrealistic. Training data-driven models requires high computational power and significant time [21]. Finally, the input-output relationship generated by a data-driven method is considered a black box, as a physical interpretation of the generated model cannot be performed [22]. This understanding is crucial to identify where a model fails. One approach that has gained interest recently consists of incorporating prior knowledge about the physics of the system into the model via data-driven methods, i.e., the so-called physics-informed machine learning (PIML). However, PIML is mainly concerned with modifying the cost function of the machine learning algorithms by incorporating the differential equations that describe the system to be modeled through the so-called physics loss, but the black-box nature of the models remains [23], [24].

Other studies have proposed hybrid models, where an LPTN of the PMSM was generated to estimate the temperature at the stator and PM, and the estimation error was compensated by cascading the estimation of the LPTN with a neural network [8], [16]. In these works, low-order LPTNs were fitted, which produced biased temperature estimates. These types of hybrid methods depend primarily on neural networks to correct the bias of the fitted LPTN, which requires a large sample size and estimation time to train them effectively [22]. The number of parameters that the proposed hybrid method requires is large, which hinders its interpretability and could tend to be overfitted if not trained with sufficient data. Moreover, the study presented in [16] did not consider the effect of the speed-dependent and temperature-dependent machine behavior.

This paper presents an in-depth extension of our previous study [25]. The authors propose and compare Hammerstein (HM) and nonlinear autoregressive exogenous models (NLARX) to estimate the temperature of the winding and PMs of an out-runner radial-flux surface-mounted PMSM, intending to achieve a target error of $\pm 5^{\circ}\mathrm{C}$. Unlike other data-driven methods, HM and NLARX models allow the incorporation of the known thermal behavior of the motor of the PMSM to be considered using a linear model. At the same time, the nonlinear function of the HM and NLARX is fitted to compensate for the residual errors of the LPTN. The linear block and function of the HM and NLARX models are generated by identifying a fourth-order LPTN of the PMSM. In contrast to our previous work, this paper accounts for the effect of the varying parameters of the LPTN by modeling the varying conduction and convection thermal resistances dependent on the coolant temperature and motor speed as temperature sources. This is performed for the initialized LPTN in the HM and NLARX models. To the best of the authors' knowledge, this is the first time HM and NLARX models have been used in combination with LPTNs to estimate the temperature of an electric machine. The main contributions of this work are listed as follows:

1) A fourth-order LPTN of an out-runner PMSM is fitted, and the varying thermal resistances dependent on the coolant temperature and the motor speed are modeled as temperature sources (inputs).
2) HM and NLARX models are used to reproduce the thermal behavior of the out-runner PMSM to estimate the temperature of the winding and PM by initializing the model via a linear time-invariant fourth-order LPTN.
3) HM and NLARX models are compared in terms of performance, number of parameters, training, and inference time for the temperature estimation in the winding and PM domains of an out-runner PMSM.

The remainder of this paper is structured as follows. Section II details the dataset and an overview of the background of the methods used in the present work. Section III presents the results of the proposed methods, while Section IV discusses them. Finally, Section V concludes the work and outlines potential future research directions.

II. Methodology

A. Dataset

The dataset used in this work was collected from a 200-kW out-runner (outer rotor) radial-flux surface-mounted PMSM with a distributed winding used for traction applications. The machine features 180 slots and 60 PM poles, an external rotor radius of 228 mm, an internal rotor radius of 196 mm and 730 mm of stack length. It can reach a maximum torque of 1.6 kNm and a maximum speed of 1.3 kr/min.

The PMSM was evaluated on a driving cycle with two pre-heating periods. Between each pre-heating period, the Federal-Test-Procedure-75 (FTP-75) and the Worldwide harmonized Light-duty Vehicles Test Cycle (WLTC) were used to drive the PMSM; see Fig. 1. High-demand driving cycles were used to ensure a dynamic thermal response of the PMSM, which is assumed to be installed on the electric powertrain of an A-class vehicle. Previous efforts have used constant torque and speed setpoints, which denote less challenging conditions for temperature estimation.

The setup used for characterization purposes is shown in Fig. 2, while a cross-section diagram of the tested motor appears in Fig. 3. A back-to-back configuration was employed, where the tested machine is mechanically coupled to an identical machine acting as a load. Winding temperature in the tested motor was measured using two glass-encapsulated thermistors (TDK/EPCOS B57551G1103). These probes were placed at the upper left and right axial end sides of the windings, corresponding to the hotspot regions. The thermistors have a resistance tolerance of $\pm 2\%$ or equivalent to $\pm 0.4^{\circ}\mathrm{C}$ in the range between -10 and $300^{\circ}\mathrm{C}$. Similarly, inlet and outlet coolant temperatures were measured using the same type of thermistors. Rotor surface temperature was measured using

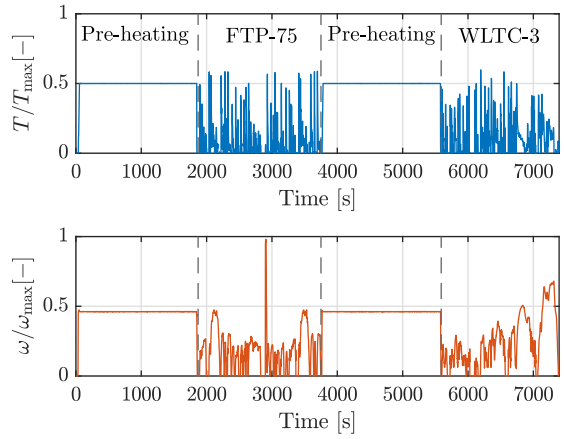

Fig. 1. Normalized driving cycle: motor torque (top) and angular speed (bottom).

a Calex PyroCouple PC151MT-0 infrared sensor with fixed emissivity (0.95), operating in the 0–250°C range. Manufacturer specifications indicate an accuracy of ±1% or ±1°C, and repeatability of ±0.5% or ±0.5°C, whichever is greater in both cases. Temperature data acquisition was carried out using a National Instruments cDAQ 9184 chassis equipped with a 9205 16-bit analog input module from the same brand.

A Kistler 4503A2K0L00B1000 torque transducer with accuracy of ±1 Nm couples mechanically the two machines. This sensor includes a built-in speed transducer. A sampling rate of 5 Hz was selected for temperature, torque, and speed signals to capture relevant dynamics with sufficient temporal resolution. In contrast, phase currents were measured through an AEMC SR 661 probe filtered at 100 kHz. Data from the inverters was captured through a CAN bus using a Dewesoft SIRIUS Digital Acquisition power analyzer at 1 kHz. Although electrical and thermal/mechanical data were stored in two acquisition setups at different sampling rates, they were carefully synchronized and resampled to 1 kHz for their use.

The average difference between the two winding temperature sensors through the driving cycle is less than 2°C. Thus, the thermistor that provides the highest temperature at the beginning of the test is used to represent the winding hotspot temperature at a single point. On the other hand, a single-point PM temperature was estimated by applying a constant offset of 5°C to the rotor surface temperature. This offset was found as the average temperature increased across multiple 3D finite-element heat transfer simulations in COMSOL Multiphysics, reproducing torque and speed working points scattered within the operating envelope of the motor. Through these simulations, the offset could be bounded between 3.5 and 6.5°C. Ansys Motor-CAD was also used as an additional FEA tool to corroborate the values obtained, as well as to generate power loss maps to be implemented in the developed models. Although the PM temperature is not directly acquired, it is referred to as *measured* in the remainder of this work because it is directly obtained from a measured signal. This step is crucial and necessary because the exact measurement of the PM temperature is difficult to implement due to the rotating nature of the PM array. Measurements during the described heating cycles were repeated for three different coolant temperatures at the inlet of the machine: 20, 40, and 60°C, and using an average coolant flow rate of 15 L/min, which was manually controlled through a valve.

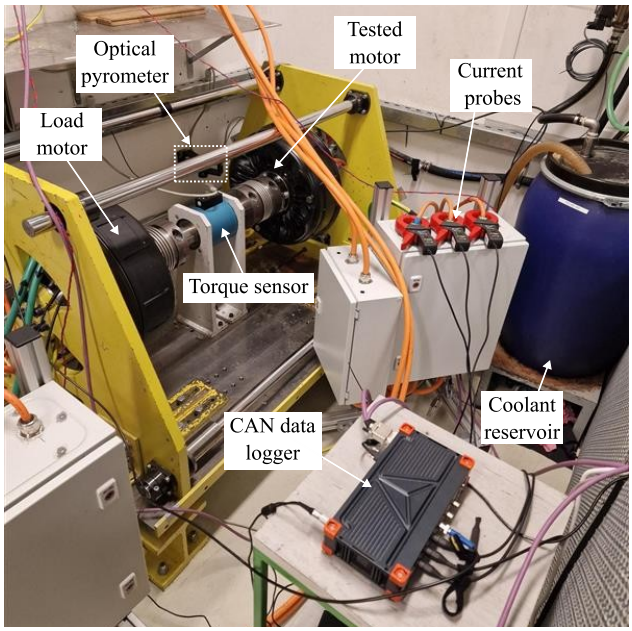

Fig. 2. In-wheel motor back-to-back test setup.

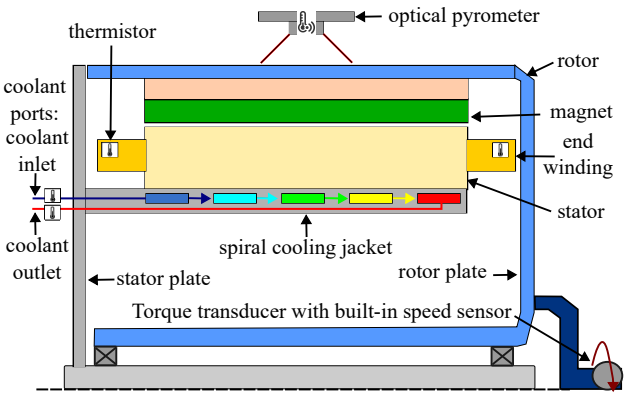

Fig. 3. Cross section of the tested in-wheel motor.

For this study, three experimental profiles were available. Due to the small sample size, a three-fold cross-validation was applied to assess the generalization performance of the fourth-order LPTN, HM, and NLARX models. For fold 1, the training set consisted of the profiles with the coolant inlet temperature of 20 and 40°C, while the validation profile was at 60°C. Folds 2 and 3 consist of the remaining combinations of inlet temperatures for training and validation. For each validation fold, the performance of the model was evaluated, and the average performance was subsequently computed based on the results from all validation folds.

B. System Identification Techniques

System identification can be performed in two ways, with a prediction focus or a simulation focus. Prediction involves estimating a model using current and past measured input and output values and initial conditions. Simulation implies that the model output is computed using only input data and initial conditions without considering measured output data. A simulation focus provides the best way to evaluate the behavior of a model under a wide range of conditions [26]. For a prediction estimation focus, the error $e_{\mathrm{p}}(t)$ represents the 1-step prediction error as shown in (1), while for a simulation estimation focus, the error $e_{\mathrm{s}}(t)$ is computed as in (2):

$$e_{\mathrm{p}}(t) = y_{\mathrm{m}}(t) - \hat{y}_{\mathrm{p}}(t) \tag{1}$$

$$e_{\mathrm{s}}(t) = y_{\mathrm{m}}(t) - \hat{y}_{\mathrm{s}}(t) \tag{2}$$

where $y_{\mathrm{m}}(t)$ is the actual measured output data of the system at time t; $\hat{y}_{\mathrm{p}}(t)$ is the prediction output value of the model at time t, and $\hat{y}_{\mathrm{s}}(t)$ is the simulated output value of the model at time t.

This work estimates the NLARX via prediction and simulation focus, while the HM models are estimated with only simulation focus via the System Identification Toolbox in MATLAB R2023b. Computations were performed on a workstation with an AMD Ryzen 9 3950X 16-core processor (3.49 GHz) and 64 GB of RAM.

C. Hammerstein Models

Hammerstein models belong to a broader class known as block-oriented models. The most popular class of block-oriented nonlinear models is the Hammerstein-Wiener model, whose general structure is illustrated in Fig. 4.

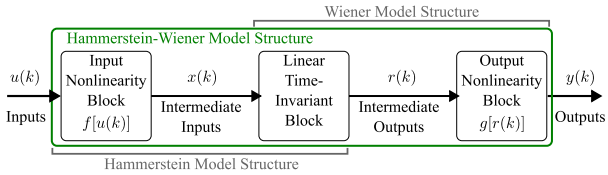

Fig. 4. General structure of a Hammerstein-Wiener model.

Hammerstein-Wiener models are characterized by three components: a linear time-invariant block representing the linear dynamics of the system to be modeled, a static input nonlinear block, and a static output nonlinear block [27]. The input and output nonlinear blocks can be modeled via artificial neural networks, a first-degree polynomial, a dead zone nonlinearity, or saturation nonlinearity. The selection of the nonlinear function depends mainly on the characteristics of the system to be modeled, or can be selected via trial and error. HM models were first described by [28] and consist of the input nonlinear block and the linear time-invariant block and can be expressed as follows:

$$\begin{aligned} x(k) =& f[u(k)], \qquad y(k) = s(k), \\ r(k) =& \frac{B(z)}{A(z)} x(k), \quad s(k) = r(k) + v(k), \end{aligned} \tag{3}$$

where $u(k)$ represents the system inputs and $y(k)$ represents the systems outputs; $x(k)$, $r(k)$ and $s(k)$ are the system intermediate variables, while $v(k)$ represents stochastic white noise with zero mean. The linear time-invariant block is represented through the polynomials $B(z)$ and $A(z)$ in the unit backward shift operator z^{-1}. The $B(z)$ polynomial can be understood as the zeros of the system, while $A(z)$ can be understood as the poles of the system. The polynomials can be defined as follows:

$$\begin{aligned} B(z) =& b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}, \\ A(z) =& 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}. \end{aligned} \tag{4}$$

HM models are frequently employed to model dynamic systems with nonlinear behavior at the inputs of the system [29]. Despite their simplicity, HM models have been used for modeling several system types, such as wind turbines, physical and chemical systems, and lithium-ion batteries [30], [31]. For this study, the linear block of the HM approach was generated by identifying a fourth-order LPTN of the PMSM. By doing this, we intend to compensate for the residual errors of the previously fitted LPTN via the input nonlinearity of the HM model.

D. Nonlinear Autoregressive Exogenous Models

Autoregressive exogenous models (ARX), in their nonlinear variety, use nonlinear functions to model a system, which helps account for the nonlinear behavior present in the dynamics of a system [32]. To understand the architecture of NLARX models, one can consider the single-input, single-output discretized form for a linear ARX model:

$$y(k) = \mathbf{a}^{\top}\mathbf{y}(k-1) + \mathbf{b}^{\top}\mathbf{u}(k) + e(k) \tag{5}$$

with

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_{\mathrm{a}}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_{\mathrm{b}}} \end{bmatrix},$$

$$\mathbf{y}(k-1) = \begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-n_{\mathrm{a}}) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(k-n_{\mathrm{b}}+1) \end{bmatrix}.$$

In (5), $\mathbf{a}$ is a vector of autoregressive coefficients, $\mathbf{b}$ is a vector of exogenous input coefficients, $\mathbf{y}(k-1)$ is a vector of past values of $y(k)$, $\mathbf{u}(k)$ is a vector of past values of $u(k)$, and $e(k)$ represents a disturbance term. Coefficients n_{a} and n_{b} represent the maximum delay of the inputs and outputs used as regressors.

On the other hand, a nonlinear ARX model allows more flexibility since instead of only considering a weighted sum of regressors that represents a linear mapping, it also considers a nonlinear mapping function F that depends on a vector $\mathbf{y}(k-1)$ composed of past values of the output $y(k)$, a vector $\mathbf{u}(k)$ that is composed of a vector of past values of the inputs $u(k)$, and a disturbance term $e(k)$ described as follows:

$$y(k) = F(\mathbf{y}(k-1), \mathbf{u}(k)) + e(k) \tag{6}$$

where

$$\mathbf{y}(k-1)=\begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-n_{\mathrm{a}}) \end{bmatrix}, \quad \mathbf{u}(k)=\begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(k-n_{\mathrm{b}}+1) \end{bmatrix}.$$

The nonlinear mapping function F could be a wavelet neural network, a sigmoid neural network, a Gaussian process regression, a support vector regression, or a regression tree. In this regard, the final prediction of a nonlinear ARX model consists of adding the output of a linear ARX model, the output of a nonlinear ARX model, and an offset term, as illustrated in Fig. 5.

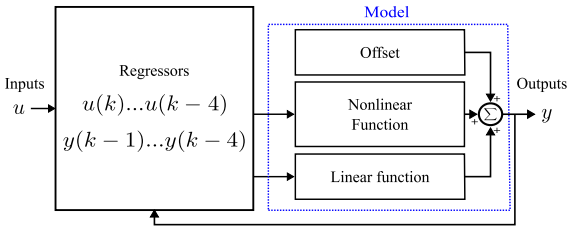

Fig. 5. NLARX model structure.

Furthermore, the process presented in Fig. 5, can be expressed as follows:

$$Y(x)=L^{\top}(x-r)+F(Q(x-r))+d \tag{7}$$

where the term $L^{\top}(x-r)$ represents the linear function block; $F(Q(x-r))$ represents the nonlinear function block, and d is an offset term. The term x represents a vector of regressors, while r is the mean of x; L and Q are matrices of parameters of the linear and nonlinear functions, respectively. If only the nonlinear function is considered, the NLARX is equivalent to the NARX models [33]. In turn, if only the linear function is considered, the NLARX model is equivalent to the linear ARX models. The latter do not account for offset terms, while NLARX models allow control of the degree of flexibility and complexity of the model, which could go from only adding a single offset term to a very deep neural network. In this study, the linear component of the NLARX model was initialized using the fourth-order LPTN of the PMSM, with the regressor delay selected according to the discrete order of the identified LPTN. As with the HM, the goal is to address the residual errors of the fitted LPTN through the nonlinear function of the NLARX structure.

E. Fourth-Order Lumped Parameter Thermal Network

The linear components of the HM and NLARX models were generated by fitting a fourth-order LPTN of the PMSM. The generated LPTN was based on the reduced order thermal circuit proposed by [14], which can be appreciated in Fig. 6a. In this case, temperature sources are equivalent to voltage sources of electrical circuits; thermal capacitances are equivalent to electrical capacitance; power losses are equivalent to current sources; and thermal resistances are akin to electrical resistance [34]. The above allows Kirchhoff voltage and current laws to be applied to the thermal circuit and generate a model that describes the heat flow of the PMSM. The inputs of the LPTN are the electric power losses in the stator and PM, as well as the ambient and coolant temperatures, while the states are the temperature in each network node.

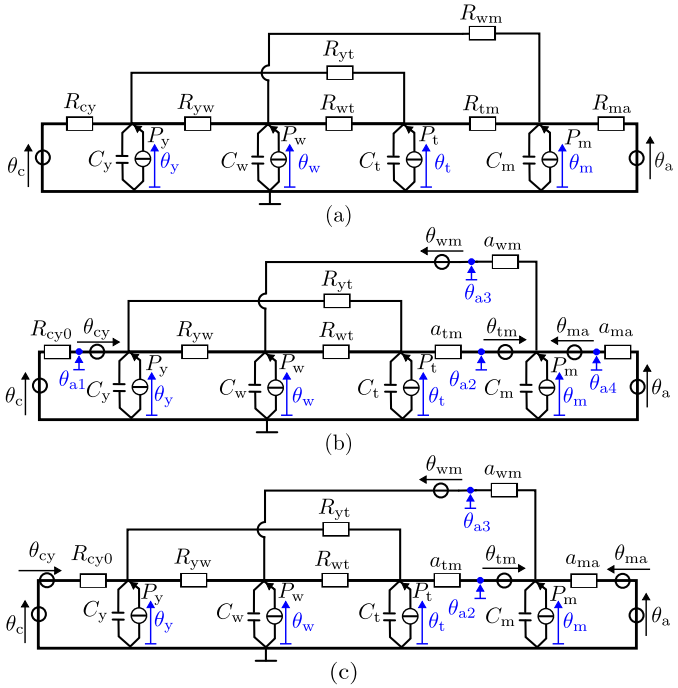

Fig. 6. Fourth order lumped-parameter thermal network (LPTN) for (a) original [14] and (b) modified model where temperature sources replace varying resistance components. A rearrangement of the temperature sources in (c) represents the varying thermal resistance between the coolant and stator yoke and between the ambient temperature and PM.

The LPTN of a PMSM is a linear parameter-varying model, as the convection thermal resistances between the PM and its surroundings depend on speed, and the conduction resistance between the coolant and stator yoke depends on coolant temperature [14]. However, the linear part of the HM and NLARX models requires a linear time-invariant system. To meet this, the speed and temperature dependent resistances were modeled as temperature sources and treated as inputs to capture their effect. A state-space form of the LPTN was then built using only fixed resistances and capacitances. The process follows the approach presented by [14].

Figs. 6b and 6c show the fourth-order LPTN with the varying thermal resistances represented via temperature sources considered in this work. The difference between Figs. 6b and 6c is that the coolant temperature sources connected to the stator yoke node and the ambient temperature sources connected to the node of the PM are rearranged since they are in series. The above was done to simplify the generation of the state-space representation of the LPTN. Despite the rearrangements presented in Fig. 6, the three thermal circuits diagrams are equivalent. The four system nodes and their subindexes are the stator yoke (y), stator winding (w), stator tooth (t), and PM (m). The external nodes are the coolant and ambient denoted with subindexes c, and a, respectively. Temperatures, heat sources, and thermal capacitances are marked

with the variables θ, P, and C, respectively. The thermal resistances are divided between convection resistances (R_{wm}, $R_{\mathrm{tm}}, R_{\mathrm{ma}}$), and conduction resistances ($R_{\mathrm{cy}}, R_{\mathrm{yw}}, R_{\mathrm{wt}}, R_{\mathrm{yt}}$). The fixed conduction resistances are R_{yw}, R_{wt}, and R_{yt}. The speed-dependent convection resistance between the PM and its surroundings (i.e., R_{wm}, $R_{\mathrm{tm}}, R_{\mathrm{ma}}$) can be described as follows according to the formulations presented in [35] and [36]:

$$R_{ij}(n) = R_{ij0} \exp\left(-\frac{\omega}{\omega_{\max}} \frac{1}{b_{ij}}\right) + a_{ij} \tag{8}$$

with

$$\begin{aligned} R_{ij0} &\in \{\mathbb{R} \mid 0 \leq R_{ij0} \leq R_{ij0,\max}\}, \\ b_{ij} &\in \{\mathbb{R} \mid 0 \leq b_{ij0} \leq b_{ij0,\max}\}, \\ a_{ij} &\in \{\mathbb{R} \mid 0 \leq a_{ij0} \leq a_{ij0,\max}\}, \end{aligned}$$

where ω is the motor speed and R_{ij0}, b_{ij}, and a_{ij} are constant parameters. Notice that the speed-dependent resistances are composed of a constant parameter and an exponential term dependent on the speed of the motor. In this regard, the exponential term of the thermal resistance was modeled with a temperature source describing the change in resistance according to the motor's speed. The constant component (a_{ij}) was placed in series to this temperature source as appreciated in Figs. 6b and 6c. The thermal resistance between the coolant and the stator yoke is [14]

$$R_{\mathrm{cy}} = R_{\mathrm{cy0}} \left[1 + \alpha_{\mathrm{cy}}(\theta_{\mathrm{c}} - \theta_{\mathrm{c0}})\right] \tag{9}$$

$$\alpha_{\mathrm{cy}} \in \{\mathbb{R} \mid -1\%/\mathrm{K} \leq \alpha_{\mathrm{cy}} \leq 0\} \tag{10}$$

where $\theta_{\mathrm{c0}} = 40°\mathrm{C}$ is the reference coolant temperature. This value was determined based on the average temperature of the coolant. Like the speed-dependent resistances, the temperature-dependent component of (9) was modeled with a source and series resistance R_{cy0} (see Figs. 6b and 6c).

Power losses in the windings, stator, and PMs are obtained from look-up tables computed through Ansys Motor-CAD for the in-wheel motor. Winding losses include AC components ($P_{\mathrm{w,ac}}$) due to proximity and skin effects [37]. Joule losses are included as well ($P_{\mathrm{w,dc}}$). The temperature dependence of both winding losses was accounted by using:

$$\begin{aligned} P_{\mathrm{w,dc}}(\theta_{\mathrm{w}}) &= P_{\mathrm{w,dc}}(1 + \alpha_{\mathrm{Cu}}(\theta_{\mathrm{w}} - \theta_{\mathrm{wr}})), \\ P_{\mathrm{w,ac}}(\theta_{\mathrm{w}}) &= P_{\mathrm{w,ac}}(1 + \alpha_{\mathrm{Cu}}(\theta_{\mathrm{w}} - \theta_{\mathrm{wr}})), \\ P_{\mathrm{w}} &= P_{\mathrm{w,dc}}(\theta_{\mathrm{w}}) + P_{\mathrm{w,ac}}(\theta_{\mathrm{w}}), \end{aligned} \tag{11}$$

where $\alpha_{\mathrm{Cu}} = 0.39\,\%/\mathrm{K}$ and $\theta_{\mathrm{wr}} = 60°\mathrm{C}$, which is the reference temperature of the winding.

Iron losses in the stator and rotor laminations are computed using the Bertotti method [38], which includes hysteresis, eddy currents, and excess loss components. Similarly, magnet Joule losses from induced eddy-currents are also considered [39].

Then, stator losses P_{s} were split into yoke and tooth components accounting for the dependence on current, $I/I_{\max}$, and speed, $\omega/\omega_{\max}$, through [14]:

$$P_{\mathrm{y}} = k_{\mathrm{s}}(I, \omega) P_{\mathrm{s}} \tag{12}$$

$$P_{\mathrm{t}} = (1 - k_{\mathrm{s}}(I, \omega)) P_{\mathrm{s}} \tag{13}$$

$$k_{\mathrm{s}}(I, \omega) = k_0 + k_1 \frac{I}{I_{\max}} + k_2 \frac{\omega}{\omega_{\max}} + k_3 \frac{I}{I_{\max}} \frac{\omega}{\omega_{\max}} \tag{14}$$

The impact of temperature on iron losses was neglected since it is not as relevant as in winding losses. Moreover, there is uncertainty in predicting stator yoke and stator tooth temperatures, as they were not measured. The total electric losses P are shown in a normalized shape in the torque-speed plane in Fig. 7. They are the summation of the individual components:

$$P = P_{\mathrm{y}} + P_{\mathrm{t}} + P_{\mathrm{w}} + P_{\mathrm{m}} \tag{15}$$

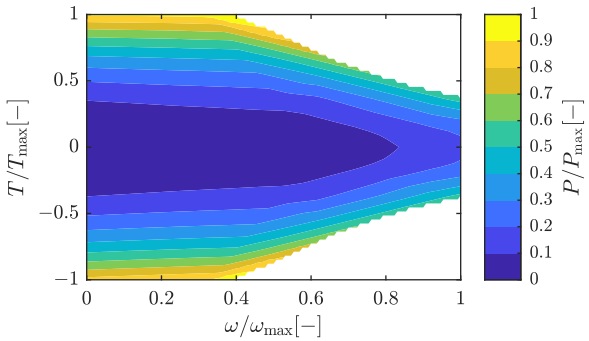

Fig. 7. Normalized electric losses in the motor torque-speed plane obtained from finite-element analysis.

Considering the above, one may generate a model of the thermal behavior of the motor by considering the ordinary differential equation that describes the energy balance at each node i of the LPTN. Heat transfer is governed by:

$$C_i \frac{\mathrm{d}\theta_i(t)}{\mathrm{d}t} = \sum_{j \neq i} \frac{\theta_j(t) - \theta_i(t)}{R_{ij}} + P_i(t) \tag{16}$$

where the left-hand side represents the rate of change of thermal energy stored in node i. The summation term accounts for the heat transfer between node i and all other nodes j in the system. $P_i(t)$ represents the external heat input (or generation) to node i at time t. Expression (16) is applied to each node in the thermal network presented in Fig. 6. The resulting system can be written in state-space form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{17}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{18}$$

The state-space matrices are defined in (26). The term G_{ij} represents the thermal conductance between nodes i and j. The state-space model was generated using the LPTN diagram in Fig. 6c. This state-space representation allows the matrix $\mathbf{A}$ to have full rank. Notice that the inputs of the LPTN are composed of the power losses in the stator and PM, the ambient and coolant temperatures, and the auxiliary temperature sources.

The auxiliary temperatures θ_{a1} to θ_{a4}—as indicated in 6b—were computed and employed as inputs to account for the effect of the varying resistances of the LPTN. The former accounts for the temperature-dependent resistance R_{cy}, while the rest are used to consider the effect of the speed-dependent resistances. In this regard, the auxiliary temperatures are

computed using the Kirchhoff voltage law in the node of interest resulting in the following expressions:

$$\begin{aligned}\theta_{\mathrm{a1}} &= \theta_{\mathrm{y}} - \theta_{\mathrm{cy}},\\ \theta_{\mathrm{a2}} &= \theta_{\mathrm{m}} - \theta_{\mathrm{tm}},\\ \theta_{\mathrm{a3}} &= \theta_{\mathrm{w}} - \theta_{\mathrm{wm}},\\ \theta_{\mathrm{a4}} &= \theta_{\mathrm{m}} - \theta_{\mathrm{ma}}.\end{aligned} \tag{19}$$

Additionally, the varying temperature sources are defined as

$$\begin{aligned}\theta_{\mathrm{wm}} &= R_{\mathrm{wm1}} P_{\mathrm{wm}},\\ \theta_{\mathrm{tm}} &= R_{\mathrm{tm1}} P_{\mathrm{tm}},\\ \theta_{\mathrm{ma}} &= R_{\mathrm{ma1}} P_{\mathrm{ma}},\\ \theta_{\mathrm{cy}} &= R_{\mathrm{cy1}} P_{\mathrm{cy}},\end{aligned} \tag{20}$$

where R_{wm1}, R_{tm1}, and R_{ma1} are the speed-dependent resistance components defined previously in (8). Lastly, the heat fluxes are computed using the constant component of the varying thermal resistances by using the expressions in (21).

$$\begin{aligned}P_{\mathrm{cy}} &= R_{\mathrm{cy0}}^{-1}(\theta_{\mathrm{a1}} - \theta_{\mathrm{c}}),\\ P_{\mathrm{tm}} &= a_{\mathrm{tm}}^{-1}(\theta_{\mathrm{a2}} - \theta_{\mathrm{t}}),\\ P_{\mathrm{wm}} &= a_{\mathrm{wm}}^{-1}(\theta_{\mathrm{a3}} - \theta_{\mathrm{m}}),\\ P_{\mathrm{ma}} &= a_{\mathrm{ma}}^{-1}(\theta_{\mathrm{a4}} - \theta_{\mathrm{a}}).\end{aligned} \tag{21}$$

The final relation of the auxiliary temperatures can be obtained by substituting (20) and (21) into (19) and solving for each auxiliary temperature:

$$\begin{aligned}\theta_{\mathrm{a1}} &= \left(\theta_{\mathrm{y}} + R_{\mathrm{cy01}} R_{\mathrm{cy0}}^{-1}\theta_{\mathrm{c}}\right)\left(1 + R_{\mathrm{cy01}} R_{\mathrm{cy0}}^{-1}\right)^{-1},\\ \theta_{\mathrm{a2}} &= \left(\theta_{\mathrm{m}} + R_{\mathrm{tm1}} a_{\mathrm{tm}}^{-1}\theta_{\mathrm{t}}\right)\left(1 + R_{\mathrm{tm1}} a_{\mathrm{tm}}^{-1}\right)^{-1},\\ \theta_{\mathrm{a3}} &= \left(\theta_{\mathrm{w}} + R_{\mathrm{wm1}} a_{\mathrm{wm}}^{-1}\theta_{\mathrm{m}}\right)\left(1 + R_{\mathrm{wm1}} a_{\mathrm{wm}}^{-1}\right)^{-1},\\ \theta_{\mathrm{a4}} &= \left(\theta_{\mathrm{m}} + R_{\mathrm{ma1}} a_{\mathrm{ma}}^{-1}\theta_{\mathrm{a}}\right)\left(1 + R_{\mathrm{ma1}} a_{\mathrm{ma}}^{-1}\right)^{-1}.\end{aligned} \tag{22}$$

Furthermore, a relationship to compute the varying temperature sources in terms of the resistance and temperature without requiring the heat flux value can be defined by substituting expressions (19) and (21) in (20) and solving for each varying temperature source:

$$\begin{aligned}\theta_{\mathrm{wm}} &= R_{\mathrm{wm1}} a_{\mathrm{wm}}^{-1}\left(1 + R_{\mathrm{wm1}} a_{\mathrm{wm}}^{-1}\right)^{-1}\left(\theta_{\mathrm{w}} - \theta_{\mathrm{m}}\right),\\ \theta_{\mathrm{tm}} &= R_{\mathrm{tm1}} a_{\mathrm{tm}}^{-1}\left(1 + R_{\mathrm{tm1}} a_{\mathrm{tm}}^{-1}\right)^{-1}\left(\theta_{\mathrm{m}} - \theta_{\mathrm{t}}\right),\\ \theta_{\mathrm{ma}} &= R_{\mathrm{ma1}} a_{\mathrm{ma}}^{-1}\left(1 + R_{\mathrm{ma1}} a_{\mathrm{ma}}^{-1}\right)^{-1}\left(\theta_{\mathrm{m}} - \theta_{\mathrm{a}}\right),\\ \theta_{\mathrm{cy}} &= R_{\mathrm{cy01}} R_{\mathrm{csy0}}^{-1}\left(1 + R_{\mathrm{cy01}} R_{\mathrm{csy0}}^{-1}\right)^{-1}\left(\theta_{\mathrm{y}} - \theta_{\mathrm{c}}\right).\end{aligned} \tag{23}$$

Based on the rearrangement of the thermal network presented in Fig. 6c, the sources for the coolant temperature θ_{c} and the varying temperature θ_{cy} are in series. The same happens for the ambient temperature θ_{a} and the varying temperature source θ_{ma}. These components can be added together, respectively, as single temperature inputs for the state-space model in (26):

$$\begin{aligned}\theta_{\mathrm{cv}} &= \theta_{\mathrm{c}} + \theta_{\mathrm{cy}},\\ \theta_{\mathrm{av}} &= \theta_{\mathrm{a}} + \theta_{\mathrm{ma}},\end{aligned} \tag{24}$$

Considering the relations provided above for the auxiliary temperatures, the effect of speed-dependent and temperature-dependent thermal resistances can be considered in the LPTN. This defines a linear time-invariant state-space representation to initialize the NLARX and HM models.

F. Identification of the Lumped Parameter Thermal Network

Following the procedure presented in [14], the fourth-order LPTN parameters were optimized using particle swarm optimization (PSO) with the MATLAB Global Optimization Toolbox. PSO is a population-based stochastic optimization algorithm inspired by the intelligent collective behavior observed in animals, such as bird flocks or fish schools [40]. Here, it is noticed that the original linear parameter-varying model was used; thus, the coefficients for speed and temperature-dependent resistances were also fitted. The initial parameters were estimated using the material weight and its properties for the capacitances, and the thermal resistances are computed from the analytical equations for electrical machines in [41] and [42]. The boundaries of the optimization were set to cover the initial estimation.

The fourth-order LPTN presented in Fig. 6 was simulated in Simulink and the simulation output was compared to that of the actual measured data to evaluate the error of the LPTN parameters. The simulation considered a sampling time of 2 seconds; thus, all sensor data was resampled to that sampling frequency.

The optimization objective was the error between the measured ($\mathbf{y}$) and estimated temperatures ($\hat{\mathbf{y}}$) for the winding and PM: $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$. Since no outliers were found in the temperature measurements and the data follows the expected temperature dynamics, no preprocessing steps such as filtering were applied before model training. The maximum likelihood cost function $J(\mathbf{p})$ was used to determine the optimal parameters in its form for multiple-input and multiple-output state-space models as given below:

$$J(\mathbf{p}) = \det\left(\sum_{i=1}^{n} \mathbf{e}_i(\mathbf{p})\mathbf{e}_i(\mathbf{p})^{\top}\right) \tag{25}$$

where n is the number of samples and $\mathbf{p}$ is the parameter vector.

Figure 8 shows the evolution of the cost function during the LPTN identification, obtained by minimizing (25) using PSO. The number of iterations was fixed to limit estimation time, with each iteration evaluating a swarm of 220 individuals. Evaluations were parallelized across sixteen Simulink-MATLAB instances to accelerate computation. The maximum number of iterations was selected by analyzing the cost function trend, which showed no further reduction beyond iteration 80. Hence, estimation was stopped at iteration 100. The cost function converged to similar values for folds 1 and 3. Table I lists the identified LPTN parameter values for each cross-validation fold.

After identifying the LPTN model parameters, the system was transformed to the time-invariant representation of Fig. 6c and expressed in a state-space form (26). Then, the model was discretized through a zero-order hold with a sampling time of 2 seconds. Discretization is crucial to initiating and training the HM and NLARX models via the fourth-order LPTN to yield a stable model.

$$\mathbf{A} = \begin{bmatrix} \frac{1}{C_\mathrm{y}} & 0 & 0 & 0 \\ 0 & \frac{1}{C_\mathrm{w}} & 0 & 0 \\ 0 & 0 & \frac{1}{C_\mathrm{t}} & 0 \\ 0 & 0 & 0 & \frac{1}{C_\mathrm{m}} \end{bmatrix} \begin{bmatrix} -\left(G_\mathrm{yw}+G_\mathrm{yt}+G_\mathrm{cy0}\right) & G_\mathrm{yw} & G_\mathrm{yt} & 0 \\ G_\mathrm{yw} & -\left(G_\mathrm{yw}+G_\mathrm{wt}\right) & G_\mathrm{wt} & \frac{1}{a_\mathrm{wm}} \\ G_\mathrm{yt} & G_\mathrm{wt} & -\left(G_\mathrm{yt}+G_\mathrm{wt}+\frac{1}{a_\mathrm{tm}}\right) & 0 \\ 0 & 0 & \frac{1}{a_\mathrm{tm}} & -\left(\frac{1}{a_\mathrm{wm}}+\frac{1}{a_\mathrm{ma}}\right) \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{C_\mathrm{y}} & 0 & 0 & 0 \\ 0 & \frac{1}{C_\mathrm{w}} & 0 & 0 \\ 0 & 0 & \frac{1}{C_\mathrm{t}} & 0 \\ 0 & 0 & 0 & \frac{1}{C_\mathrm{m}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{R_\mathrm{cy0}} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{a_\mathrm{wm}} \\ 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{a_\mathrm{tm}} & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{1}{a_\mathrm{ma}} & -\frac{1}{a_\mathrm{tm}} & \frac{1}{a_\mathrm{wm}} \end{bmatrix}, \quad \mathbf{C} = \mathbf{I}_{4\times 4}, \quad \mathbf{D} = \mathbf{0}_{4\times 8},$$

$$\mathbf{x} = \begin{bmatrix} \theta_\mathrm{y} & \theta_\mathrm{w} & \theta_\mathrm{t} & \theta_\mathrm{m} \end{bmatrix}^\top, \quad \mathbf{u} = \begin{bmatrix} P_\mathrm{y} & P_\mathrm{w} & P_\mathrm{t} & P_\mathrm{m} & \theta_\mathrm{cv} & \theta_\mathrm{av} & \theta_\mathrm{a2} & \theta_\mathrm{a3} \end{bmatrix}^\top. \tag{26}$$

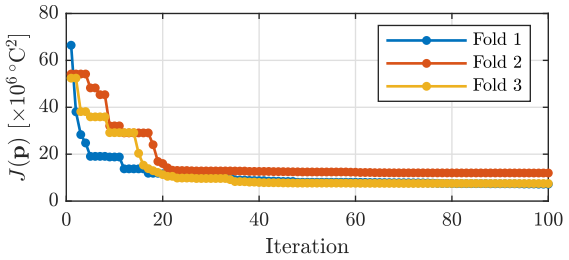

Fig. 8. Cost function minimization during the parameter identification process for the fourth-order LPTN. Results for each cross-validation fold are reported.

TABLE I
IDENTIFIED PARAMETER VALUES PER FOLD

Parameter	Unit	Fold 1	Fold 2	Fold 3
C_m	J/K	6846	6652	6834
C_t		1167	3043	3222
C_w		5738	5296	5533
C_y		1649	3340	2979
$R_\mathrm{cy,0}$	K/W	0.005	0.004	0.004
$R_\mathrm{tm,0}$		0.408	0.484	0.402
$R_\mathrm{wm,0}$		0.682	0.665	0.579
$R_\mathrm{w,t}$		0.522	0.436	0.362
$R_\mathrm{y,t}$		0.015	0.015	0.016
$R_\mathrm{y,w}$		0.009	0.009	0.009
$R_\mathrm{ma,0}$		0.305	0.314	0.291
a_tm		0.078	0.094	0.071
a_wm		0.418	0.492	0.499
a_ma		0.032	0.036	0.028
b_ma	–	0.187	0.176	0.204
b_tm		0.356	0.200	0.248
b_wm		1.198	0.968	0.804
k_0	–	0.697	0.775	0.800
k_1		−0.085	−0.080	−0.092
k_2		−0.008	−0.008	−0.005
k_3		−0.159	−0.115	−0.165
α_cy	1/K	−0.009	−0.009	−0.008

G. Identification of the Hammerstein Model

The HM modeling approach was used to estimate the temperature of the PM and winding. To this end, the identified LPTN model was used to model the linear block of the HM model. Since the HM approach requires linear time-invariant systems to be fitted, the model in (26) was employed. In contrast, the input nonlinear functions used for the HM model were one-neuron sigmoid networks characterized by the function:

$$S(X) = \sum_{i=1}^{n} s_i f\left(X^\top Q b_i + c_i\right) \tag{27}$$

where $S(X)$ is the output of the sigmoid network; X is the input vector; n is the number of sigmoid units (neurons); s_i are scalar weights, called output coefficients; Q is an $m \times q$ projection matrix, where $m \geq q$; b_i are dilation coefficient vectors, each of size $q \times 1$; c_i are scalar translations, and $f(z) = [1 + \exp(-z)]^{-1}$ is the sigmoid function, where $z = X^\top Q b_i + c_i$.

The one-neuron sigmoid networks used as the input nonlinearity were applied to the inputs related to the power losses of the LPTN model, while the temperature sources were fed directly to the LPTN model. Neural networks were used for the static input nonlinearity of the HM model since they are considered universal approximators of functions [16]. One-neuron neural networks were selected to keep the number of parameters of the HM model low. Although a higher number of neurons could improve the performance of the HM model, it could lead to overfitting. The inputs related to temperature sources were directly fed to the LPTN without being processed via a nonlinearity. Figure 9 illustrates the LPTN integration into the HM model structure.

The HM model was trained with the Levenberg-Marquardt (LM) algorithm. The LM algorithm is an optimization technique for solving nonlinear least squares problems. It combines gradient descent and Gauss-Newton methods [43]. The maximum number of iterations to fit the HM model was 100, and mean normalization was performed on the inputs to which the input nonlinearity was applied for estimating the model. In addition, the eigenvalues of the LPTN were fixed while estimating the HM model. Figure 10 shows the cost function of the HM model. After 40 iterations, there is no significant reduction in the cost function.

H. Identification of the NLARX model

Similarly to the HM model, the previously identified LPTN model was used to initialize the linear function of the NLARX model. Since the LPTN is a fourth-order model, four regressors were used to fit the NLARX model for each input and

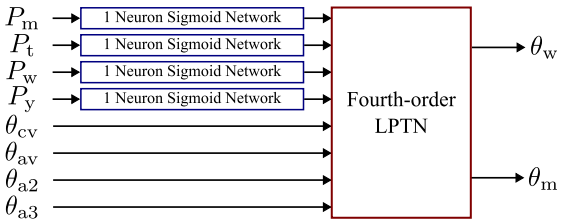

Fig. 9. HM model structure initialized with the fourth-order LPTN of the PMSM.

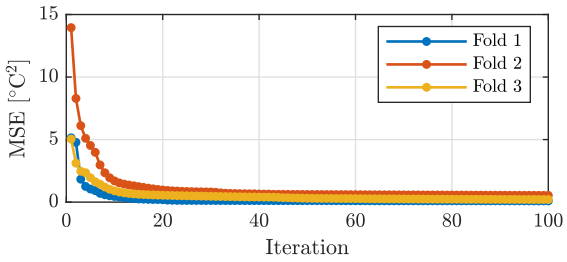

Fig. 10. Cost function minimization during the parameter identification process for the HM model with a simulation focus. Results for each cross-validation fold are reported.

output. The above implies that the number of lag values for the inputs u and the outputs y go from $k-1$ to $k-4$. The above led to a total of 40 regressors being considered, 32 associated with the inputs of the linear time-invariant form of the LPTN (see (26)) and eight associated with the outputs i.e., the PM and winding temperature. On the other hand, the nonlinear function used was a neural network of one layer and one neuron with a sigmoid activation function as in (27). This neural network was used to maintain a low number of parameters for the NLARX model and avoid overfitting. The structure of the NLARX model used in this work is shown in Fig. 11.

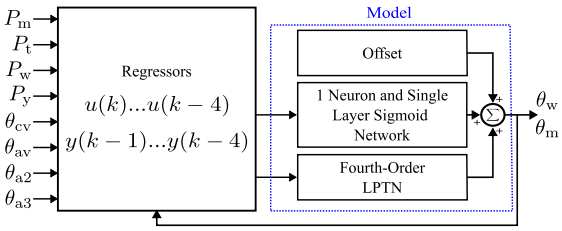

Fig. 11. NLARX model structure initialized with the fourth-order LPTN of the PMSM.

The NLARX model was estimated using the LM algorithm over 15 iterations. Furthermore, input and output data were normalized using mean normalization before estimating the NLARX model. The model was fitted with a simulation focus, with the LPTN zeros remaining fixed during estimation. MATLAB 2023b implements NLARX estimation in two steps. First, it initializes the parameters by minimizing the prediction error. Then, it refines the model with a simulation focus to enhance the estimation. Figure 12a shows the cost function evolution using a prediction focus, while Fig. 12b shows the simulation-focused results.

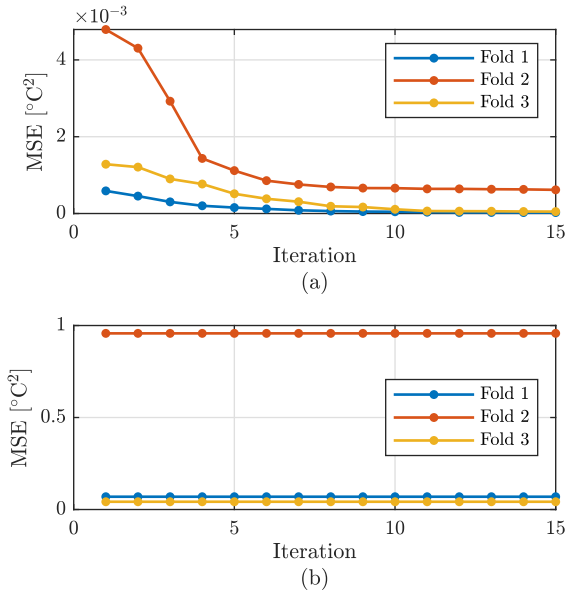

Fig. 12. Cost function minimization during the parameter identification process for the NLARX model with (a) prediction focus and (b) simulation focus. Results for each cross-validation fold are reported.

Notice that the cost function values with prediction focus are lower than those with simulation focus. This is because, in prediction focus, the actual past output values are used to predict the next output, while in simulation focus, the model relies on its previous output, introducing an error depending on its accuracy. Additionally, the cost function starts below one, and the improvement across 15 iterations is small when compared to the HM model cost function behavior.

III. Results

To assess the performance of the LPTN, HM and NLARX models, the instantaneous error is defined for each sample as

$$e_i = y_i - \hat{y}_i \tag{28}$$

where y_i is the ith temperature observation, and $\hat{y}_i$ is its estimated value. Specific metrics were extracted from the instantaneous error i.e., mean-squared-error (MSE), root-mean-squared error (RMSE), and mean absolute error (MAE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \tag{29}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{30}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{31}$$

being n the number of data points. Additionally, the R-squared value was also computed to assess how much of the variation in the dependent variable can be explained by the proposed models:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{32}$$

with $\bar{y}$ representing the mean of the measured values.

The infinity norm error was used to measure the largest absolute difference between the predicted and actual values [44]. It is defined as the maximum absolute error in the dataset and can be computed as follows:

$$\|e\|_\infty = \max_i |e_i| \tag{33}$$

Finally, the normalized root-mean-squared error (NRMSE) was used to measure the performance of the fitted models. The NRMSE is a statistical metric used to calculate the differences between predicted and observed values, normalized to make the comparison dimensionless. The NRMSE can be computed as follows:

$$\text{NRMSE} = \frac{\text{RMSE}}{\sigma_y} = \frac{\sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{34}$$

These metrics were selected to evaluate the fitted models holistically and align with those reported by [14], [18], [19], [45]. Table II shows the validation performance based on these metrics for the fourth-order LPTN, HM, and NLARX models across each cross-validation fold. The best performing metrics for winding and PM temperature estimations are marked with an asterisk.

TABLE II
VALIDATION PERFORMANCE METRICS FOR LPTN, HM, AND NLARX MODELS

Metric	LPTN		HM		NLARX	
	Wdg.	PM	Wdg.	PM	Wdg.	PM
Fold 1						
MSE [°C^2]	12.78	1.165*	3.953	1.280	3.144*	1.384
RMSE [°C]	3.575	1.080*	1.988	1.132	1.773*	1.177
MAE [°C]	3.135	0.851*	1.742	0.936	1.554*	1.087
$\|e\|_\infty$ [°C]	6.476	2.725	5.871	3.146	3.796*	2.628*
R^2	0.985	0.946*	0.995	0.941	0.996*	0.936
NRMSE	0.123	0.232*	0.069	0.243	0.061*	0.253
Fold 2						
MSE [°C^2]	10.28	0.632	1.400*	0.479	3.234	0.396*
RMSE [°C]	3.207	0.795	1.183*	0.692	1.798	0.629*
MAE [°C]	3.055	0.692	1.015*	0.572	1.548	0.555*
$\|e\|_\infty$ [°C]	5.328	1.775	4.513*	2.628	5.887	1.347*
R^2	0.988	0.975	0.998*	0.981	0.996	0.984*
NRMSE	0.111	0.159	0.041*	0.138	0.063	0.126*
Fold 3						
MSE [°C^2]	8.669	0.293	0.630	0.330	0.492*	0.081*
RMSE [°C]	2.944	0.541	0.793	0.574	0.702*	0.285*
MAE [°C]	2.660	0.434	0.592	0.452	0.527*	0.221*
$\|e\|_\infty$ [°C]	5.232	1.463	3.410	2.371	2.279*	1.073*
R^2	0.990	0.987	0.999	0.985	0.999*	0.996*
NRMSE	0.102	0.116	0.027	0.123	0.024*	0.061*

*Best performance

Table III shows the average performance and standard deviation for each model computed from the validation folds in Table II. Again, the best metrics across methods are marked with asterisks. Figures 13 and 14 compare the LPTN, HM, and NLARX models for winding and PM temperatures, respectively, using the validation set from fold 2. The top plot shows the measured temperature and model responses, while the bottom plot displays the error over time. Initial conditions were set using the compare function from the System Identification Toolbox in MATLAB R2023b.

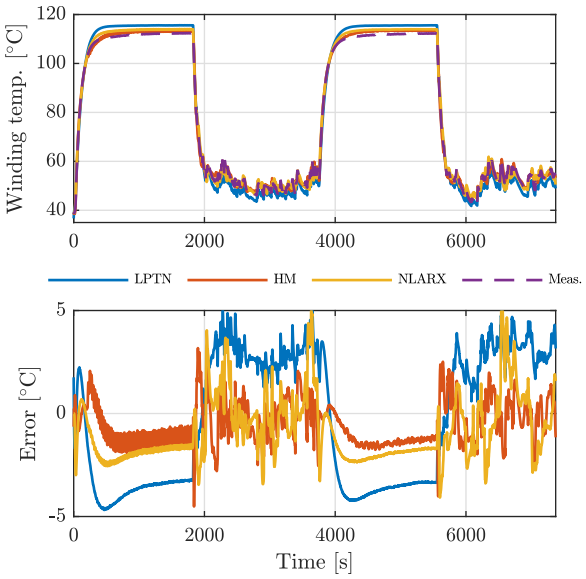

Fig. 13. Winding temperature estimation comparison between LPTN, HM, and NLARX models.

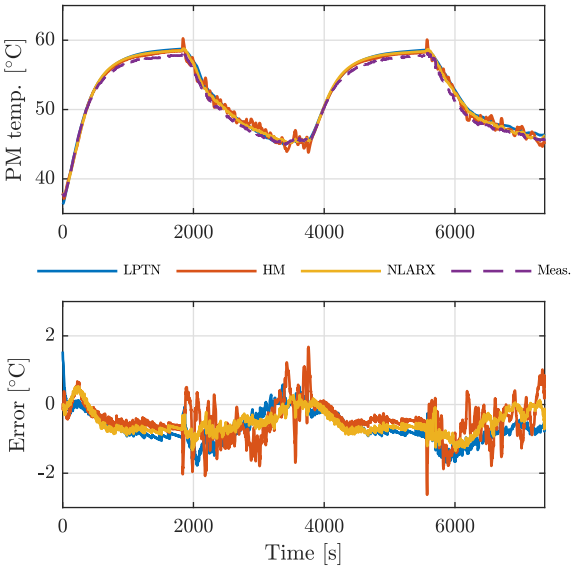

Fig. 14. PM temperature estimation comparison between LPTN, HM, and NLARX models.

The number of parameters of each model was computed to compare their complexity, as reported in Table IV. Additionally, the average training time of the LPTN, HM, and NLARX models is reported to compare the computational power needed to estimate each of them. To evaluate the inference time of the proposed models in real-time hardware, a Processor-in-the-Loop (PIL) simulation was performed using MATLAB R2023b. The discrete models were implemented with a 2-second sampling time and zero-order hold. The simulation ran on a C2000™ LaunchPad with a TMS320F28069M microcontroller operating at 90 MHz, tailored for motor control

TABLE III
AVERAGE VALIDATION PERFORMANCE AND STANDARD DEVIATION FOR EACH METRIC AND MODEL

Metric	LPTN				HM				NLARX			
	Winding		PM		Winding		PM		Winding		PM	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
MSE [$^\circ$C^2]	10.58	2.072	0.697	0.440	1.994*	1.739*	0.696	0.511	2.290	1.558	0.621*	0.680*
RMSE [$^\circ$C]	3.242	0.317	0.805	0.269	1.322*	0.609*	0.799	0.294	1.424	0.626	0.697	0.450*
MAE [$^\circ$C]	2.950	0.255	0.659	0.210	1.116*	0.582*	0.653	0.252	1.210	0.591	0.6212*	0.4366*
$\|e\|_\infty$ [$^\circ$C]	5.679	0.692	1.988	0.657	4.598	1.233	2.715	0.395	3.987*	1.812*	1.683*	0.830*
R^2	0.987	0.003	0.969	0.021	0.998*	0.002*	0.969	0.024	0.997	0.002	0.972*	0.032*
NRMSE	0.112	0.012	0.169	0.059	0.046*	0.021*	0.168	0.065	0.049	0.022	0.147*	0.098*

*Best performance

applications. The device features 256 KB of flash memory and 100 KB of RAM. These results are also listed in Table IV.

TABLE IV
TRAINING AND PIL RESULTS FOR LPTN, HM, AND NLARX MODELS

Model	Parameters	Avg. Training Time [s]	Std. Dev. Training [s]	Execution Time [μs]	CPU Util. [%]
LPTN	22	$46 \cdot 10^3$	209	60.7	0.003
HM	136	23	3.8	161	0.008
NLARX	150	44	42.5	1417	0.071

IV. DISCUSSION

Table II shows that both HM and NLARX reduce winding temperature estimation error across all folds compared to the fourth-order LPTN. In folds one and three, the HM model slightly increases PM error, but this is offset by the greater reduction in winding error. Moreover, the HM consistently yields R-squared values closer to one for winding temperature estimation than the LPTN.

Despite fold-to-fold variation, Table III shows that the average error of the NLARX and HM models is lower than that of the fourth-order LPTN for both winding and PM temperature estimation. The HM performs better for winding, while NLARX gives a lower PM error. The limited PM improvement, slightly below that of the LPTN, may result from the out-runner PMSM design, where the PM is more exposed to windage and less affected by heat transfer from the winding (i.e., the main heat source) [46]. The main improvement of the HM and NLARX approaches lies in reducing the winding temperature estimation error, while the improvement in PM temperature estimation is limited. It is worth noting that both models are initialized with the previously identified LPTN, with the proposed nonlinearities aiming to reduce the residual error of the LPTN. Notably, their PM RMSE, MAE, and maximum error remain within $\pm 5^\circ$C.

From Figs. 13 and 14, the HM model shows lower error for the winding throughout the entire profile compared to the LPTN. For PM temperature, the HM shows slight improvements during specific intervals but exhibits higher error in several instances, particularly in fold 2, where its maximum absolute error exceeds that of the LPTN (see Table II). However, for MSE, RMSE, MAE, NRMSE, and R-squared, the HM model performs better, particularly in winding temperature estimation. The NLARX model also exhibits lower error than the LPTN during pre-heating and driving cycle periods for winding temperature. For PM temperature estimation, both models' responses closely match the reference values. Still, NLARX achieves lower error across all metrics (see Table II) and slightly higher R-squared for both winding and PM. On average (Table III), NLARX performs marginally better than the fourth-order LPTN for PM estimation.

Model complexity analysis (Table IV) shows that HM and NLARX consist of 136 parameters and greater, respectively, with NLARX being more complex, though both remain simpler than gradient boosting trees, CNNs, and LSTMs, which typically exceed one thousand parameters [8], [45], [47]. NLARX requires longer training due to its feedback-based nonlinearity, unlike the HM's static nonlinearities. Despite fewer estimation iterations, NLARX's simulation-focused training increases computational time. In contrast, training the fourth-order LPTN via PSO with speed and temperature-dependent parameters took over 12 hours for 100 iterations. Since both HM and NLARX are initialized from the discretized LPTN, estimation time remains low (under 60 seconds). PIL results show that execution time is lowest for the LPTN, while NLARX has the highest, though all models consume less than 0.1% CPU and fit within the available flash memory.

Despite the lower error and higher R-squared performance of both the HM and NLARX models, the nonlinear functions introduced by these methods result in more complex models compared to the fourth-order LPTN, which only involves 22 parameters. Therefore, it is necessary to evaluate whether this improvement persists under wider conditions or in real time and justifies the increased complexity. Nonetheless, the number of parameters introduced by the proposed methods remains lower than those reported by [8], [15], [20] when using neural networks and low-order LPTNs.

Alternative nonlinearities for the proposed NLARX and HM models could be explored. This study used single-neuron, single-layer sigmoid networks, but architectures such as wavelet or deep neural networks may yield improved performance within these frameworks. However, increased network complexity risks overfitting, requiring larger datasets and regularization. This study did not include comparisons with deep models like LSTMs or CNNs, which demand extensive data and introduce significant computational overhead [48], and no clear guidelines exist to define the required sample size to train such models reliably. These considerations motivated the use of single-neuron sigmoid networks in both approaches.

Using an LPTN to initialize the linear component of the NLARX and HM models requires computing power losses,

commonly calculated through FEA. However, acquiring accurate torque measurements outside the test bench is challenging due to the lack of precise torque loading and sensing. Power losses could also be approximated by the HM model input nonlinearity using speed and current instead of look-up tables generated via FEA. To compensate potential temperature inaccuracies due to the open-loop response of the LPTN, HM, and NLARX models, feedback approaches like precise flux observers and Kalman filters could be used [49]. Additionally, setting adequate initial conditions in real time is not addressed here but could be explored in future work, as discussed in [50]. Hence, the real-time implementation of these models remains an open question.

V. Conclusion and Future Work

This work proposed the use of HM and NLARX models to estimate the winding and PM temperatures of an outrunner PMSM. Both models were initialized with a previously identified fourth-order LPTN of the PMSM and compared. The speed-dependent and temperature-dependent thermal resistances of the LPTN were modeled using temperature sources and used as inputs to account for their effects on a time-invariant representation of the LPTN. The results show that the LPTN achieved an average MSE of $10.578^{\circ}\mathrm{C}^2$ for winding temperature estimation and $0.697^{\circ}\mathrm{C}^2$ for PM temperature estimation. In comparison, the HM model achieved an MSE of $1.994^{\circ}\mathrm{C}^2$ for winding temperature and $0.696^{\circ}\mathrm{C}^2$ for PM temperature. On the other hand, the NLARX model achieved an MSE of $2.290^{\circ}\mathrm{C}^2$ for winding temperature and $0.621^{\circ}\mathrm{C}^2$ for PM temperature. The obtained values are deemed acceptable since the corresponding RMSE values are within the $\pm 5^{\circ}\mathrm{C}$ target error.

Between the HM and NLARX models, the NLARX offered a slight improvement in PM temperature estimation, while the HM significantly outperformed in winding temperature estimation. Future work could explore integrating higher or lower-order LPTNs with HM or NLARX models for PMSM temperature estimation. Testing these methods on other motor types and operating conditions remains underexplored. Lastly, further studies should assess the feasibility of real-time implementation.

VI. Acknowledgments

The authors would like to thank Elaphe Propulsion Technologies Ltd. for providing the experimental data used in the development of the models. The work herein was developed during the EM-TECH research project part of the 2Zero partnership funded by the European Commission (Grant agreement ID: 101096083). Erick Axel Martinez-Ríos and Irving S. Aguilar-Zamorate are funded by Consejo Nacional de Humanidades, Ciencias y Tecnologias (CONAHCYT) (scholarships 1010770 and 1105077, respectively). Saulius Pakštys is funded by the PNRR-NGEU project receiving funding from MUR – DM 352/2022.

References

[1] J.-G. Lee and D.-K. Lim, "A stepwise optimal design applied to an interior permanent magnet synchronous motor for electric vehicle traction applications," *IEEE Access*, vol. 9, p. 115090–115099, 2021.

[2] T. Orlowska-Kowalska, M. Wolkiewicz, P. Pietrzak, M. Skowron, P. Ewert, G. Tarchala, M. Krzysztofiak, and C. T. Kowalski, "Fault diagnosis and fault-tolerant control of pmsm drives–state of the art and future challenges," *IEEE Access*, vol. 10, p. 59979–60024, 2022.

[3] Y. Park, D. Fernandez, S. B. Lee, D. Hyun, M. Jeong, S. K. Kommuri, C. Cho, D. Diaz Reigosa, and F. Briz, "Online detection of rotor eccentricity and demagnetization faults in PMSMs based on hall-effect field sensor measurements," *IEEE Trans. Ind. Appl.*, vol. 55, no. 3, pp. 2499–2509, May 2019.

[4] D. D. Reigosa, D. Fernandez, T. Tanimoto, T. Kato, and F. Briz, "Permanent-magnet temperature distribution estimation in permanent-magnet synchronous machines using back electromotive force harmonics," *IEEE Trans. Ind. Appl.*, vol. 52, no. 4, pp. 3093–3103, Jul. 2016.

[5] G. Turabee, M. R. Khowja, P. Giangrande, V. Madonna, G. Cosma, G. Vakil, C. Gerada, and M. Galea, "The role of neural networks in predicting the thermal life of electrical machines," *IEEE Access*, vol. 8, pp. 40 283–40 297, 2020.

[6] A. Gandhi, T. Corrigan, and L. Parsa, "Recent advances in modeling and online detection of stator interturn faults in electrical motors," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, p. 1564–1575, May 2011.

[7] O. Wallscheid, "Thermal monitoring of electric motors: State-of-the-art review and future challenges," *IEEE Open J. Ind. Applicat.*, vol. 2, pp. 204–223, 2021.

[8] Z. Liu, W. Kong, X. Fan, Z. Li, K. Peng, and R. Qu, "Hybrid thermal modeling with lptn-informed neural network for multi-node temperature estimation in pmsm," *IEEE Transactions on Power Electronics*, p. 1–12, 2024.

[9] A. Boglietti, A. Cavagnino, D. Staton, M. Shanel, M. Mueller, and C. Mejuto, "Evolution and modern approaches for thermal analysis of electrical machines," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, p. 871–882, Mar. 2009.

[10] Z. Song, R. Huang, W. Wang, S. Liu, and C. Liu, "An improved dual iterative transient thermal network model for pmsm with natural air cooling," *IEEE Transactions on Energy Conversion*, vol. 37, no. 4, p. 2588–2600, Dec. 2022.

[11] O. Wallscheid, T. Huber, W. Peters, and J. Böcker, "A critical review of techniques to determine the magnet temperature of permanent magnet synchronous motors under real-time conditions," *EPE Journal*, vol. 26, no. 1, p. 11–20, Jan. 2016.

[12] L. He, Y. Feng, Y. Zhang, and B. Tong, "Methods for temperature estimation and monitoring of permanent magnet: a technology review and future trends," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 4, Mar. 2024.

[13] S. D. Wilson, P. Stewart, and B. P. Taylor, "Methods of resistance estimation in permanent magnet synchronous motors for real-time thermal management," *IEEE Transactions on Energy Conversion*, vol. 25, no. 3, p. 698–707, Sep. 2010.

[14] O. Wallscheid and J. Bocker, "Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors," *IEEE Transactions on Energy Conversion*, vol. 31, no. 1, p. 354–365, Mar. 2016.

[15] W. Kirchgassner, O. Wallscheid, and J. Bocker, "Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark," *IEEE Transactions on Energy Conversion*, vol. 36, no. 3, p. 2059–2067, Sep. 2021.

[16] L. Jin, Y. Mao, X. Wang, L. Lu, and Z. Wang, "A model-based and data-driven integrated temperature estimation method for pmsm," *IEEE Transactions on Power Electronics*, vol. 39, no. 7, p. 8553–8561, Jul. 2024.

[17] O. Wallscheid, W. Kirchgassner, and J. Bocker, "Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017.

[18] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105537, Jan. 2023.

[19] J. Lee and J.-I. Ha, "Temperature estimation of pmsm using a difference-estimating feedforward neural network," *IEEE Access*, vol. 8, p. 130855–130865, 2020.

[20] W. Kirchgassner, O. Wallscheid, and J. Bocker, "Estimating electric motor temperatures with deep residual machine learning," *IEEE Transactions on Power Electronics*, vol. 36, no. 7, p. 7480–7488, Jul. 2021.
[21] ——, "Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors," in *2019 IEEE International Electric Machines & Drives Conference (IEMDC)*. IEEE, May 2019.
[22] 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, p. 188–214, 2023.
[23] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, p. 686–707, Feb. 2019.
[24] P. Wang, X. Wang, and Y. Wang, "Physics-informed machine learning based permanent magnet synchronous motor temperature estimation," in *2023 9th International Conference on Mechanical and Electronics Engineering (ICMEE)*. IEEE, Nov. 2023.
[25] E. A. Martinez-Ríos, I. S. Aguilar-Zamorate, S. Pakštys, R. Galluzzi, and N. Amati, "Hammerstein models for rotor and winding temperature estimation of a permanent magnet synchronous motor," in *2024 International Symposium on Electromobility (ISEM)*, 2024, pp. 1–8.
[26] L. Ljung, *System Identification*. Birkhäuser Boston, 1998, p. 163–173.
[27] M. Schoukens and K. Tiels, "Identification of block-oriented nonlinear systems starting from linear approximations: A survey," *Automatica*, vol. 85, p. 272–292, Nov. 2017.
[28] A. Hammerstein, "Nichtlineare integralgleichungen nebst anwendungen," *Acta Mathematica*, vol. 54, no. 0, p. 117–176, 1930.
[29] F. Li and L. Jia, "Parameter estimation of hammerstein–wiener nonlinear system with noise using special test signals," *Neurocomputing*, vol. 344, p. 37–48, Jun. 2019.
[30] X. Yue, Q. Fu, H. Hu, and D. Wang, "Hammerstein model based battery soc estimation considering temperature variation," *Journal of The Electrochemical Society*, vol. 171, no. 3, p. 030503, mar 2024.
[31] Y. Xu, L. Jia, D. Peng, and W. Yang, "Iterative neuro-fuzzy hammerstein model based model predictive control for wind turbines," *IEEE Transactions on Industry Applications*, vol. 59, no. 5, p. 6501–6512, Sep. 2023.
[32] L. Ljung, Q. Zhang, P. Lindskog, A. Iouditski, and R. Singh, "An integrated system identification toolbox for linear and non-linear models," *IFAC Proc. Vol.*, vol. 39, no. 1, pp. 931–936, 2006.
[33] M. A. Lopez-Carmona and A. Paricio Garcia, "Linear and nonlinear model predictive control (mpc) for regulating pedestrian flows with discrete speed instructions," *Physica A: Statistical Mechanics and its Applications*, vol. 625, p. 128996, Sep. 2023.
[34] G. D. Demetriades, H. Z. de la Parra, E. Andersson, and H. Olsson, "A real-time thermal model of a permanent-magnet synchronous motor," *IEEE Transactions on Power Electronics*, vol. 25, no. 2, p. 463–474, Feb. 2010.
[35] D. A. Howey, P. R. N. Childs, and A. S. Holmes, "Air-gap convection in rotating electrical machines," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, p. 1367–1375, Mar. 2012.
[36] D. A. Staton and A. Cavagnino, "Convection heat transfer and flow calculations suitable for electric machines thermal models," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 10, p. 3509–3516, Oct. 2008.
[37] P. Mellor, R. Wrobel, and N. Simpson, "AC losses in high frequency electrical machine windings formed from large section conductors," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2014, pp. 5563–5570.
[38] G. Bertotti, "General properties of power losses in soft ferromagnetic materials," *IEEE Transactions on Magnetics*, vol. 24, pp. 621–630, 1988.
[39] P. Zhang, G. Y. Sizov, J. He, D. M. Ionel, and N. A. O. Demerdash, "Calculation of Magnet Losses in Concentrated-Winding Permanent-Magnet Synchronous Machines Using a Computationally Efficient Finite-Element Method," *IEEE Transactions on Industry Applications*, vol. 49, 2013.
[40] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, p. 387–408, Jan. 2017.
[41] A. Boglietti, A. Cavagnino, M. Lazzari, and A. Pastorelli, "A simplified thermal model for variable speed self cooled industrial induction motor," in *Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No.02CH37344)*, vol. 2, Oct. 2002.
[42] P. Mellor, D. Roberts, and D. Turner, "Lumped parameter thermal model for electrical machines of TEFC design," *IEE Proceedings B Electric Power Applications*, vol. 138, p. 205, 1991.
[43] B. M. Wilamowski and H. Yu, "Improved computation for levenberg–marquardt training," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, p. 930–937, Jun. 2010.
[44] M. Hladík and M. Černý, "Total least squares and chebyshev norm," *Procedia Computer Science*, vol. 51, p. 1791–1800, 2015.
[45] H. Jing, Z. Chen, X. Wang, X. Wang, L. Ge, G. Fang, and D. Xiao, "Gradient boosting decision tree for rotor temperature estimation in permanent magnet synchronous motors," *IEEE Transactions on Power Electronics*, vol. 38, no. 9, p. 10617–10622, Sep. 2023.
[46] Y. Chen, P. Pillay, and A. Khan, "Pm wind generator comparison of different topologies," in *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, vol. 3. IEEE, 2004, p. 1405–1412.
[47] E. G. Gedlu, O. Wallscheid, and J. Bocker, "Temperature estimation of electric machines using a hybrid model of feed-forward neural and low-order lumped-parameter thermal networks," in *2021 IEEE International Electric Machines & Drives Conference (IEMDC)*. IEEE, May 2021.
[48] H. Li, G. Pinto, M. S. Piscitelli, A. Capozzoli, and T. Hong, "Building thermal dynamics modeling with deep transfer learning using a large residential smart thermostat dataset," *Engineering Applications of Artificial Intelligence*, vol. 130, p. 107701, Apr. 2024.
[49] D. E. G. Erazo, O. Wallscheid, and J. Böcker, "Improved fusion of permanent magnet temperature estimation techniques for synchronous motors using a kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 1708–1717, 2020.
[50] D. Liang, Z. Q. Zhu, J. Feng, Y. Li, and S. Guo, "Initial temperature determination for real-time thermal models in permanent magnet synchronous machines," *IEEE Transactions on Transportation Electrification*, pp. 1–1, 2024.

Erick Axel Martinez-Ríos was born in Mexico City, Mexico, in May 1995. He received the B.Sc. degree (Hons.) in mechatronics engineering and the M.Sc. degree (Hons.) in engineering science from Tecnologico de Monterrey, Mexico City, in 2017 and 2022, respectively. He is currently pursuing the Ph.D. degree in engineering science with Tecnologico de Monterrey. From 2018 to 2020, he worked as a project specialist at the same institution. He is the author of eleven journal articles, four conference papers, and one book chapter. His research interests include signal processing, machine learning, and control theory.

Irving S. Aguilar-Zamorate received the B.Sc. in mechanical engineering and the M.S. degree in Engineering science from Tecnologico de Monterrey, Mexico City, in 2020 and 2022, respectively. He is currently pursuing a Ph.D. degree in engineering science at the same university with a research stay at the Mechatronics Laboratory, Politecnico di Torino. His research interests include thermal management, temperature estimation, design, and control of electric machines for automotive and micro-mobility applications.

Saulius Pakštys received both B.Sc. and M.Sc. degrees in mechanical engineering from Politecnico di Torino, Turin, Italy in 2021 and 2022 respectively. He is currently pursuing a Ph.D. degree in mechanical engineering at the Mechatronics Laboratory at the same institution. His research interests include energy management strategies for hybrid energy storage systems, alternative automotive powertrains, highly integrated battery systems design and their thermal management, and electric machine thermal analysis.

Renato Galluzzi (M'21–SM'23) received the M.Sc. and Ph.D. degrees in mechatronics from Politecnico di Torino, Italy, in 2010 and 2014, respectively. Since 2011, he has been an active collaborator of the Mechatronics Laboratory at Politecnico di Torino. He is currently a Research Professor in the School of Engineering and Sciences, Tecnologico de Monterrey, Mexico. He is the author of numerous journal publications and conference papers. He also holds inventorship in multiple international patents. His research interests include vibration control and damping systems, power actuators, electric machinery, and energy harvesting. Dr. Galluzzi is a member of the National Research Fellows System level 1 (SNII-1), CONAHCyT, Mexico.

Nicola Amati received the Ph.D. degree in machine design from the Department of Mechanical Engineering at Politecnico di Torino, Turin, Italy, in 2001. He joined the faculty of Politecnico di Torino in the same year, where he is now a full professor in the Mechanical and Aerospace Engineering Department. He is also a principal investigator in the Center of Automotive Research and Sustainable Mobility from the same institution. His research and lecturing interests focus on the analysis, design, and control of electromechanical systems with an emphasis on rotating machinery, active and passive magnetic bearings and dampers, and more-electric systems for automotive applications.