Received 16 November 2023; revised 12 January 2024; accepted 17 January 2024. Date of publication 22 January 2024; date of current version 5 February 2024. The review of this paper was arranged by Associate Editor Giampaolo Buticchi.

Digital Object Identifier 10.1109/OJIES.2024.3356721

Multiobjective Hyperparameter Optimization of Artificial Neural Networks for Optimal Feedforward Torque Control of Synchronous Machines

NIKLAS MONZEN , FLORIAN STROEBL , HERBERT PALM ,
AND CHRISTOPH M. HACKL (Senior Member, IEEE)

Institute for Sustainable Energy Systems (ISES), Hochschule München University of Applied Sciences, 80333 Munich, Germany

CORRESPONDING AUTHOR: CHRISTOPH M. HACKL (e-mail: christoph.hackl@hm.edu).

This work was supported by the Munich University of Applied Sciences HM and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Projectnumber 512819356.

ABSTRACT Multiobjective hyperparameter optimization is applied to find optimal artificial neural network (ANN) architectures used for optimal feedforward torque control (OFTC) of synchronous machines. The proposed framework allows to systematically identify Pareto optimal ANNs with respect to multiple (partly) contradictory objectives, such as approximation accuracy and computational burden of the considered ANNs. The obtained Pareto optimal ANNs are trained and implemented on a realtime system and tested experimentally for a nonlinear reluctance synchronous machine against non-Pareto optimal ANN designs and a state-of-the-art OFTC approach. Finally, based on the most recent results from ANN approximation theory, guidelines for Pareto optimal ANN-based OFTC design and implementation are provided.

INDEX TERMS Artificial neural network (ANN), hyperspace exploration (HSE), multiobjective hyperparameter optimization (MO-HPO), optimal feedforward torque control (OFTC), reluctance synchronous machine (RSM).

NOMENCLATURE

$\mathbb{N}, \mathbb{R}$: natural, real numbers.

$\boldsymbol{x} := (x_1, \dots, x_n)^\top \in \mathbb{R}^n$: column vector, $n \in \mathbb{N}$ where “$\top$” and “$:=$” mean “transposed” and “is defined as”, respectively.

$\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^\top \boldsymbol{x}} = \sqrt{x_1^2 + \cdots + x_n^2}$: Euclidean norm of $\boldsymbol{x}$.

$X \in \mathbb{R}^{n\times m}$: matrix with n rows and m columns.

$I_n \in \mathbb{R}^{n\times n} := \mathrm{diag}(1, \dots, 1)$: identity matrix.

$T_\mathrm{p}(\phi_\mathrm{p}) = \begin{bmatrix} \cos(\phi_\mathrm{p}) & -\sin(\phi_\mathrm{p}) \\ \sin(\phi_\mathrm{p}) & \cos(\phi_\mathrm{p}) \end{bmatrix}$: Park transformation matrix with angle $\phi_\mathrm{p} \in \mathbb{R}$ and $J := T_\mathrm{p}(\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$: counter-clock wise rotation matrix. “s.t.”: subject to (optimization with constraints).

$\mathbb{D}, \mathbb{U}, \mathbb{T}$: design, use case and target space, respectively.

n_j: number of neurons in j-th layer. $\boldsymbol{x} := (x_1, \dots, x_{n_0})^\top \in \mathbb{R}^{n_0}$: input vector to ANN.

m: number of hidden layers.

$\widehat{\boldsymbol{y}} := (\widehat{y}_{m+1,1}, \dots, \widehat{y}_{m+1,n_{m+1}})^\top \in \mathbb{R}^{n_{m+1}}$: output vector of ANN.

$\Phi_{j,i}$: activation function (AF) of ith neuron in jth layer.

$\boldsymbol{\Phi}_j := (\Phi_{j,1}, \dots, \Phi_{j,n_j})^\top$: jth layer AF vector.

I. INTRODUCTION

Electrical drives are used in various applications. While conventional control algorithms are still in use, machine learning algorithms become more and more popular (see detailed literature reviews in [1], [2]). Artificial neural networks (ANNs) are used, e.g., for control (see [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] to name a few) or for state and parameter estimation (see, e.g., [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]). The main goals of ANN-based algorithms are 1) increasing accuracy, 2) improving robustness, and 3) reducing computational effort based on data-driven approaches in contrast to physics-based approaches (although both can and should be combined).

In [2] and [28], optimal feedforward torque control (OFTC) is realized by ANNs. OFTC plays a crucial role in almost all applications as it guarantees optimal control of the electrical drives within its physical constraints [29], [30]. The core idea of OFTC is to feed *optimal* reference currents to the underlying current controllers. By determining the *optimal* reference currents, the electrical drive system produces the reference torque (or at least the maximally feasible torque), while keeping losses to a minimum and adhering to voltage and current limits.

OFTC of synchronous machine (SM) is currently an actively researched and widely investigated topic: in particular of permanent magnet synchronous machine (PMSM) and reluctance synchronous machine (RSM) (see e.g., [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]) and of electrically excited synchronous machine (EESM; see, e.g., [49], [50], [51], [52], [53], [54], [55], [56], [57], [58]). But only [48] considers and allows for 1) arbitrary nonlinearities (e.g., saturation, (d, q)-cross-coupling and cross-coupling between stator and exciter), 2) iron and copper losses, 3) stator and exciter voltage and current limits, and 4) an analytical computation for a realtime implementation. As it is an iterative algorithm (in the sense of sequential quadratic programming including online linearization), the convergence rate and, therefore, the computation time is not deterministic and must be limited to ensure realtime implementation.

To overcome this drawback, in [2] and [28] ANN-based OFTC is proposed and its efficacy is demonstrated. Both papers also highlight the importance of choosing an *optimal* ANN architecture (including the number of layers and neurons per layer) to achieve *best* performance. However, both papers follow a predominantly heuristic trial-and-error-based approach for ANN design. The most effective and efficient design of ANNs is still an open research question across a variety of disciplines [59], [60], [61], [62], [63], [64], [65].

For electric drives, ANN-based approaches [8], [19], [21], [66], [67], [68], [69] generally do not follow a systematically derived ANN architecture. In [2], [3], [4], [5], [6], [7], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [23], [24], [25], [27], [28], [70], the ANN architectures for drive control were chosen *heuristically*. Whereas [10] determined the ANN architecture by *trial and error*. In [70] and [27], the number of neurons per hidden layer was increased until the error decreased below a certain threshold or the number of neurons per hidden layer was decreased until the error increased above a certain value again, respectively. Recently, in [2] and [28] for ANN-based OFTC, it was shown, that the number of neurons per hidden layer does *not* (necessarily) correlate with the ANN's accuracy. In [28], the *optimal* ANN architecture was designed based on a (simple) hyperparameter study. However, solely 200 architectures of nearly 1.3 million possible architectures were considered.

In [26], an evolution programming method was used to determine the number of neurons per hidden layer. But it neither describes the target criteria (such as accuracy or execution time) nor the algorithm itself in detail. Moreover, it is unclear whether an *optimal* ANN architecture was found.

To the best knowledge of the authors, there is no generic, structural and a priori method (yet) to determine ANN architectures (neither the number of hidden layers nor the numbers of neurons per hidden layer) for ANN-based approaches in the field of electrical drives.

This article proposes a novel and complete framework that allows to systematically identify a (Pareto-)*optimal* ANN architecture with respect to multiple (partly) contradictory objectives by using multi-objective hyperparameter optimization (MO-HPO). Furthermore, the proposed framework is consistent with the most recent results from approximation theory of neural networks [71] in order to derive generic ANN design guidelines for ANN-based approaches in electrical drive systems.

Multiobjective optimization in electrical drives is widely used during machine design (see detailed literature reviews [72], [73], [74]). For drive control, multiobjective optimization is partially utilized for model predictive controller design (e.g., [75], [76], [77]) or (conventional) controller parameter tuning (e.g., [78], [79]).

For machine learning in general, multiobjective optimization has already been used. [80] gives a comprehensive review of algorithms used for MO-HPO. The importance of MO-HPO in practical applications is emphasized due to its capability to analyze trade-offs between multiple objectives. However, there are few applications of MO-HPO to solve ANN-based problems. One notable example is the MO-HPO-approach proposed by Parsa et al. in [81] to perform hardware/software co-optimization with respect to ANN performance (accuracy) and energy consumption.

In conclusion, to the best knowledge of the authors, this is the very first time that MO-HPO for ANN-based control or estimation approaches in electrical drive systems (in particular, for ANN-based OFTC) is proposed. The novel framework provides the opportunity to design realtime applicable ANNs for electrical drive systems and allows for their implementation on standard realtime interfaces considering the admissible computation load [e.g., floating point operations (FLOPS)]. The framework's primary goal is to identify *Pareto optimal* ANNs that strike a balance between accuracy and execution time all stemming and exploiting the most recent results from approximation theory of neural network architectures [71]. As a result, design guidelines are proposed that reduce the design space significantly. The contributions of this article are 1) generic dataset generation for the supervised learning of ANNs, 2) MO-HPO of ANNs to find Pareto optimal designs, 3) rationalization of MO-HPO results using approximation theory, and 4) experimental validation of optimally chosen ANNs for ANN-based OFTC.

The rest of this article is organized as follows. Section II introduces the problem statement and proposed solution for

MO-HPO of ANN architectures in electrical drive systems. Section III applies the proposed MO-HPO to ANN-based OFTC of nonlinear RSMs. Section III-A describes the underlying nonlinear optimization problem to create the required data set(s) for optimization and training. Section III-B discusses ANN architecture selection and ANN training. The proposed multiobjective ANN-hyperparameter optimization is discussed in Section III-C. Section III-D presents the MO-HPO results which allow to determine Pareto optimal ANN designs. These ANN designs are implemented on a realtime system and tested experimentally against non-Pareto optimal ANN designs and a state-of-the-art approach. Section IV discusses these results and, based on approximation theory, provides guidelines for optimal ANN-based OFTC design and implementation. Finally, Section V concludes this article.

II. PROBLEM STATEMENT AND PROPOSED SOLUTION

This section presents a brief discussion of the supervised ANN learning, focusing on the relationship between input and output data sets, and the parameterization of ANNs. It then delves into the problem of finding optimal ANN designs for electric drives, considering multiple objectives, such as accuracy and computational complexity. Finally, it introduces a novel workflow that employs MO-HPO to identify Pareto optimal ANN designs suitable for real-time applications in electrical drives.

A. SUPERVISED LEARNING OF ARTIFICIAL NEURAL NETWORKS

For supervised learning of ANNs, the relation between input $\boldsymbol{x}$ and output $\boldsymbol{y}$ is given by several sample data sets $(\boldsymbol{y}[k]; \boldsymbol{x}[k])$ (e.g., obtained by measurements or simulations). All $k \in \{1, \ldots, K\}$ sample sets are collected in the dataset

$$(\boldsymbol{Y}; \boldsymbol{X}) := ((\boldsymbol{y}[1]; \boldsymbol{x}[1]), \ \ldots, (\boldsymbol{y}[K]; \boldsymbol{x}[K])) \tag{1}$$

containing all corresponding output $\boldsymbol{y}[k]$ and input $\boldsymbol{x}[k]$ vectors. The output $\boldsymbol{y}[k]$ shall be approximated by an ANN, i.e.,

$$\widehat{\boldsymbol{y}}[k] = \boldsymbol{f}_{(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\Phi})}(\boldsymbol{x}[k]) \tag{2}$$

which is a recursively defined function $\boldsymbol{f}_{(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\Phi})}(\cdot)$ parameterized by ANN weights, biases, and activation functions, i.e.,

$$\left.\begin{array}{lll} \boldsymbol{w} & := \left(\boldsymbol{w}_{0,1}^\top, \ldots, \boldsymbol{w}_{m,n_{m+1}}^\top\right)^\top \in \mathbb{R}^{n_0+\sum_{k=1}^{m+1} n_{k-1}n_k} \\ \boldsymbol{b} & := \left(b_{0,1}, \ldots, b_{m,n_{m+1}}\right)^\top \in \mathbb{R}^{\sum_{k=0}^{m+1} n_k} \\ \boldsymbol{\Phi} & := \left(\Phi_{0,1}, \ldots, \Phi_{m,n_{m+1}}\right)^\top \in \mathbb{R}^{\sum_{k=0}^{m+1} n_k} \end{array}\right\} \tag{3}$$

where m and $n_0, n_1, n_2, \ldots, n_{m+1} \in \mathbb{N}$ are the number of hidden layers and the numbers of neurons per layer, respectively.

For MO-HPO, several ANN architectures must be evaluated. Each ANN architecture (with a fixed selection of activation functions, number of hidden layers, and number of neurons per layer) is trained to obtain its optimal parameters

$$(\boldsymbol{w}^*, \boldsymbol{b}^*) = \underset{(\boldsymbol{w},\boldsymbol{b})}{\arg\min}\ \underbrace{\tfrac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{y}[k] - \widehat{\boldsymbol{y}}[k]\|^2}_{=:e_\mathrm{y}}. \tag{4}$$

by minimizing the mean squared error (MSE) e_y. Various training algorithms, including gradient descent [82, Sec. 10.4.4], Levenberg–Marquardt [82, Sec. 4.1] and adaptive moment estimation (Adam) algorithm [83], can be employed for training. To ensure effective training and prevent overfitting of the ANN, the dataset is divided into training and validation subsets. The training set is used to fine-tune the ANN's weights and biases, while the validation set gauges the ANN's accuracy when dealing with previously unseen data. Training concludes when the validation error consistently rises over a specified number of consecutive assessments, known as *maximum validation failure runs* [84]. The adapted ANN parameters, obtained before the raise of the validation error, are employed as the optimal weights and biases for ANN implementation.

B. PROBLEM STATEMENT

ANN-based approaches for electric drives must satisfy multiple objectives and require simultaneous optimization to determine the set of multiobjective *optimal* ANN design alternatives. The main objectives are related to accuracy [e.g., minimizing e_y in (4)] and realtime implementation. In view of memory and storage constraints, the number of parameters

$$N_\mathrm{param} = \underbrace{\sum_{k=0}^{m+1} n_k}_{\text{number of biases}} + \underbrace{n_0 + \sum_{k=1}^{m+1} n_{k-1}n_k}_{\text{number of weights}} \tag{5}$$

including both weights and biases, should be a primary consideration in the ANN design. Furthermore, concerning the mathematical operations involved (see [28]), it is possible to calculate the number of FLOPs as

$$N_\mathrm{flop} = 2\,n_0 + 2\,n_{m+1}\,n_m + \sum_{k=1}^{m} n_k \left(2\,n_{k-1} + 1\right) \tag{6}$$

which helps to estimate the execution time of the ANN, i.e.,

$$t_\mathrm{exec} = N_\mathrm{flop}\frac{N_\mathrm{c}}{f_\mathrm{s}} \tag{7}$$

with clock frequency f_c and number N_c of clock cycles per FLOP.

The decision, whether the considered ANN is *optimal*, must take ANN architecture but also ANN training into account. The ANN design space covers 1) vector $\boldsymbol{\Phi}$ of chosen activation functions, 2) number m of hidden layers, 3) numbers $n_1, n_2, \ldots, n_m$ of neurons per hidden layer, 4) training method, and 5) its training parameters.

In conclusion, in view of ANN-based OFTC, obtaining *optimal* ANNs defined by their architecture and training represents the trade-off between accuracy and computational burden.

C. METHODOLOGY (PROPOSED SOLUTION)

We introduce a novel workflow that employs MO-HPO to identify *Pareto optimal* ANNs that are suitable for realtime electrical drive applications. The workflow begins with the generation of a dataset for supervised learning of ANNs. Next, the hyperspace of hyperparameters is defined which must be explored (e.g., the number m of hidden layers, the numbers $n_1, n_2, \ldots, n_m$ of neurons per hidden layer, training methods, and training parameters) and the objectives according to which we want to optimize (e.g., MSE e_{y}, number N_{flop} of FLOPs and number N_{param} of parameters). The core of the workflow lies in the execution of a MO-HPO that identifies the Pareto front efficiently and effectively by, e.g., model-based direct search algorithms [85], genetic algorithms [86], or hybrid algorithms [87], [88]. MO-HPO, along with the training of several ANNs, yields a set of Pareto optimal ANN designs that represent solutions balancing multiple objectives. These results serve as the foundation for the investigation of ANN complexity and approximation accuracy with which the following research question shall be answered: How many hidden layers and neurons are necessary for an ANN to achieve a prescribed lower bound of accuracy, while realtime implementation is still feasible? The results can be used to draw conclusions regarding the selection of future design spaces for MO-HPO (or heuristical approaches) of similar applications. During the decision-making process, we carefully navigate through the trade-offs and balance the multiobjectives to select the most suitable ANN related to the electrical drive application. Finally, the chosen ANNs are subject to validation through simulations and/or experiments showing their efficacy and implementability.

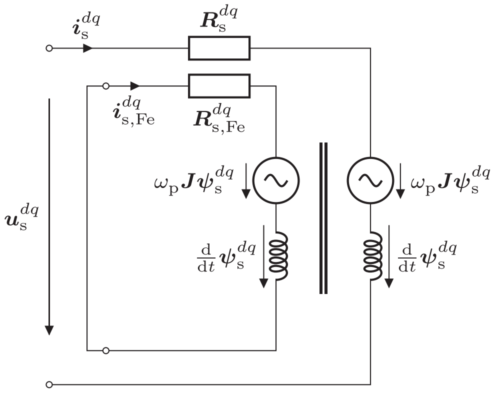
III. MULTIOBJECTIVE ANN-ARCHITECTURE OPTIMIZATION FOR ANN-BASED OFTC OF SMS

This section discusses the application of the proposed MO-HPO for ANN design for OFTC of SMs. It begins with preparations, including modeling, OFTC for RSM or (I/S)PMSM, and dataset generation. The section then covers architecture selection and training of ANNs, followed by multiobjective ANN-hyperparameter optimization. Finally, it concludes with validation, including decision making of the MO-HPO outcome and the presentation of experimental results.

A. PREPARATIONS

1) MODELING

The equivalent circuit of a nonlinear SM [48] is shown in Fig. 1. The stator and rotor iron core create a transformer-like coupling, resulting in flux linkages. Consequently, the (averaged[1]) stator flux linkages $\boldsymbol{\psi}_{\mathrm{s}}^{dq} = \boldsymbol{\psi}_{\mathrm{s}}^{dq}(\boldsymbol{i}_{\mathrm{s,m}}^{dq}, \omega_{\mathrm{p}})$, become dependent on electrical angular velocity ω_{p} and magnetization current $\boldsymbol{i}_{\mathrm{s,m}}^{dq} = \boldsymbol{i}_{\mathrm{s}}^{dq} + \boldsymbol{i}_{\mathrm{s,Fe}}^{dq}$ with stator currents $\boldsymbol{i}_{\mathrm{s}}^{dq}$ and stator iron currents $\boldsymbol{i}_{\mathrm{s,Fe}}^{dq}$. Neglecting flux leakage, stator flux linkage, and stator iron flux linkage are entirely linked, i.e., $\boldsymbol{\psi}_{\mathrm{s,Fe}}^{dq} = \boldsymbol{\psi}_{\mathrm{s}}^{dq}$, which leads to [48]



$\boldsymbol{i}_{\mathrm{s}}^{dq}$
$\boldsymbol{R}_{\mathrm{s}}^{dq}$
$\boldsymbol{i}_{\mathrm{s,Fe}}^{dq}$
$\boldsymbol{R}_{\mathrm{s,Fe}}^{dq}$
$\boldsymbol{u}_{\mathrm{s}}^{dq}$
$\omega_{\mathrm{p}}\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{s}}^{dq}$
$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq}$
$\omega_{\mathrm{p}}\boldsymbol{J}\boldsymbol{\psi}_{\mathrm{s}}^{dq}$
$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq}$

FIGURE 1. Equivalent circuit of the nonlinear SM model.

$$\left.\begin{aligned} \boldsymbol{u}_{\mathrm{s}}^{dq} &= R_{\mathrm{s}}^{dq}\boldsymbol{i}_{\mathrm{s}}^{dq} + \omega_{\mathrm{p}}J\boldsymbol{\psi}_{\mathrm{s}}^{dq} + \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq} \\ \boldsymbol{0}_2 &= R_{\mathrm{s,Fe}}^{dq}\boldsymbol{i}_{\mathrm{s,Fe}}^{dq} + \omega_{\mathrm{p}}J\boldsymbol{\psi}_{\mathrm{s}}^{dq} + \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq} \end{aligned}\right\} \tag{8}$$

with $J := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, $\boldsymbol{0}_2 := (0, 0)^\top$, stator voltages $\boldsymbol{u}_{\mathrm{s}}^{dq} := (u_{\mathrm{s}}^{d}, u_{\mathrm{s}}^{d})^\top$, stator currents $\boldsymbol{i}_{\mathrm{s}}^{dq} := (i_{\mathrm{s}}^{d}, i_{\mathrm{s}}^{d})^\top$, stator resistance matrix $R_{\mathrm{s}}^{dq} \in \mathbb{R}^{2\times 2}$, stator iron currents $\boldsymbol{i}_{\mathrm{s,Fe}}^{dq} := (i_{\mathrm{s,Fe}}^{d}, i_{\mathrm{s,Fe}}^{d})^\top$, mechanical angular velocity $\omega_{\mathrm{m}} = \frac{\omega_{\mathrm{p}}}{n_{\mathrm{p}}}$ (i.e., electrical angular velocity ω_{p} divided by pole pair number n_{p}). The stator iron resistance matrix $R_{\mathrm{s,Fe}}^{dq} := R_{\mathrm{s,Fe}}^{dq}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \omega_{\mathrm{p}}) \in \mathbb{R}^{2\times 2}$ varies with current and frequency, i.e., [48]

$$R_{\mathrm{s,Fe}}^{dq} = \frac{2}{3\kappa^2}\frac{\left(\omega_{\mathrm{p}}J\boldsymbol{\psi}_{\mathrm{s}}^{dq} + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq}\right)^\top\left(\omega_{\mathrm{p}}J\boldsymbol{\psi}_{\mathrm{s}}^{dq} + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_{\mathrm{s}}^{dq}\right)}{p_{\mathrm{Fe}}}I_2 \tag{9}$$

with $I_2 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, nonlinear iron losses $p_{\mathrm{Fe}}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \omega_{\mathrm{m}})$ and Clarke transformation factor $\kappa \in \{2/3, \sqrt{2/3}\}$ [89, Ch. 14].

Invoking the principles of energy conservation and power equilibrium in combination with (8) yields

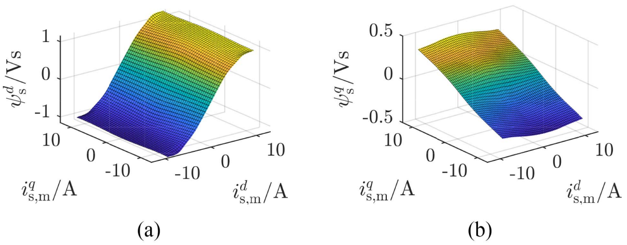
$$\left.\begin{aligned} p_{\mathrm{Cu}} &= \tfrac{2}{3\kappa^2}\left(\boldsymbol{i}_{\mathrm{s}}^{dq}\right)^\top R_{\mathrm{s}}^{dq}\boldsymbol{i}_{\mathrm{s}}^{dq}, \quad p_{\mathrm{Fe}} = \tfrac{2}{3\kappa^2}\left(\boldsymbol{i}_{\mathrm{s,Fe}}^{dq}\right)^\top R_{\mathrm{s,Fe}}^{dq}\boldsymbol{i}_{\mathrm{s,Fe}}^{dq} \\ m_{\mathrm{m}} &= \tfrac{2n_{\mathrm{p}}}{3\kappa^2}\left(\boldsymbol{i}_{\mathrm{s}}^{dq} + \boldsymbol{i}_{\mathrm{s,Fe}}^{dq}\right)^\top J\boldsymbol{\psi}_{\mathrm{s}}^{dq}, \quad p_{\mathrm{V}} = p_{\mathrm{Cu}} + p_{\mathrm{Fe}} \end{aligned}\right\} \tag{10}$$

with stator copper losses p_{Cu}, collective nonlinear machine losses $p_{\mathrm{V}}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \omega_{\mathrm{m}})$, and nonlinear machine torque $m_{\mathrm{m}}(\boldsymbol{i}_{\mathrm{s}}^{dq}, \omega_{\mathrm{m}})$. Note that, in motor mode, the stator iron currents become negative reducing the torque due to iron losses; in generator mode, the opposite is true.

[1] Neglecting angle / position dependence of the flux linkages.

TABLE 1. Key Parameters of the Investigated RSM

Parameter	Symbol	Value
Rated mech. power	$p_{\mathrm{m,R}}$	$1.9\,\mathrm{kW}$
Pole pairs	n_{p}	2
Rated mech. speed	$\omega_{\mathrm{m,R}}$	$157\,\mathrm{rad/s}$
Rated mech. torque	$m_{\mathrm{m,R}}$	$12\,\mathrm{Nm}$
Rated voltage	$\widehat{u}_{\mathrm{s,R}}$	$310.3\,\mathrm{V}$
Rated current	$\widehat{\imath}_{\mathrm{s,R}}$	$7\,\mathrm{A}$
Rated d apparent inductance	$L^{d}_{\mathrm{s,app}}$	$185\,\mathrm{mH}$
Rated q apparent inductance	$L^{d}_{\mathrm{s,app}}$	$33\,\mathrm{mH}$
Stator resistance	R_{s}	$1.3\,\Omega$



(a) (b)

FIGURE 2. Illustration of the nonlinear flux linkages ψ^d_{s} (a) and ψ^q_{s} (b) of the investigated RSM.

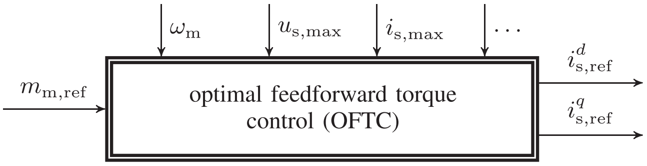


FIGURE 3. *Optimal* reference currents depend on machine constraints and actual operating conditions.

The machine under investigation is a nonlinear RSM with a power rating of 1.9 kW. All machine parameters are collected in Table 1. The nonlinear averaged flux linkage maps (obtained from measurements) are shown in Fig. 2.

2) OFTC FOR RSM OR (I/S)PMSM

As shown in Fig. 3, OFTC computes the optimal reference currents (see, e.g., [46])

$$\boldsymbol{i}^{dq}_{\mathrm{s,ref}} := \begin{pmatrix} i^d_{\mathrm{s,ref}} \\ i^q_{\mathrm{s,ref}} \end{pmatrix} := \begin{pmatrix} i^d_{\mathrm{s,ref}}(m_{\mathrm{m,ref}}, \omega_{\mathrm{m}}, u_{\mathrm{s,max}}, i_{\mathrm{s,max}}) \\ i^q_{\mathrm{s,ref}}(m_{\mathrm{m,ref}}, \omega_{\mathrm{m}}, u_{\mathrm{s,max}}, i_{\mathrm{s,max}}) \end{pmatrix} \tag{11}$$

where $i^d_{\mathrm{s,ref}}$ and $i^q_{\mathrm{s,ref}}$ are the stator (d, q)-reference currents. The reference currents are functions of reference torque $m_{\mathrm{m,ref}}$, mechanical angular velocity $\omega_{\mathrm{m}} = \frac{\omega_{\mathrm{p}}}{n_{\mathrm{p}}}$, stator constraints $u_{\mathrm{s,max}}$ and $i_{\mathrm{s,max}}$ (maximum voltage and current magnitude, respectively). The main goals of OFTC are 1) satisfaction of voltage and current constraints, 2) generation of the reference (or, at least, the maximally feasible) machine torque, and 3) minimization of (iron and copper) losses.

An operation within voltage and current limits is ensured if

$$\|\boldsymbol{u}^{dq}_{\mathrm{s}}(\boldsymbol{i}^{dq}_{\mathrm{s}})\| \le u_{\mathrm{s,max}} \quad \text{and} \quad \|\boldsymbol{i}^{dq}_{\mathrm{s}}\| \le i_{\mathrm{s,max}} \tag{12}$$

hold. The optimal and feasible torque can be attained by minimizing the deviation between reference torque $m_{\mathrm{m,ref}}$ and actual machine torque m_{m}. Efficient and loss-minimal operation is guaranteed by minimizing all machine losses, denoted as p_{V}, which includes copper p_{Cu} and iron p_{Fe} losses.

3) DATASET GENERATION

The optimal reference currents are the solution of the nonlinear optimization problem (NLP)

$$\begin{aligned} \left(\boldsymbol{i}^{dq\top}_{\mathrm{s}}, s\right)^{\top} &= \underset{\left({\boldsymbol{i}^{dq}_{\mathrm{s,ref}}}^{\top}, s^*\right)^{\top}}{\arg\min} \; p_L(\boldsymbol{i}^{dq}_{\mathrm{s}}) + h\,s \\ s.t. \quad & \|\boldsymbol{u}^{dq}_{\mathrm{s}}(\boldsymbol{i}^{dq}_{\mathrm{s}})\| \le u_{\mathrm{s,max}} \\ & \|\boldsymbol{i}^{dq}_{\mathrm{s}}\| \le i_{\mathrm{s,max}} \\ & |m_{\mathrm{m,ref}} - m_{\mathrm{m}}(\boldsymbol{i}^{dq}_{\mathrm{s}})| \le s. \end{aligned} \tag{13}$$

The NLP aims to reduce the losses while ensuring an operation within the voltage and current constraints. The parameter h must be chosen such that $p_{\mathrm{V}}(\boldsymbol{i}^{dq}_{\mathrm{s}}) \ll h\,s$ even for very small deviation $|m_{\mathrm{m,ref}} - m_{\mathrm{m}}(\boldsymbol{i}^{dq}_{\mathrm{s}})|$; i.e., the slack variable $s \ge 0$ has the lowest possible value and therefore assures the provision of the maximally feasible reference torque.

A dataset [recall (1)] is produced by solving the NLP (e.g., by MATLAB's function `fmincon`) iteratively for $k \in \{1, \ldots, K\}$ sample sets $(\boldsymbol{y}[k]; \boldsymbol{x}[k])$ with input vector

$$\boldsymbol{x}[k] := N_{\mathrm{x}} \left(m_{\mathrm{m,ref}}[k], \omega_{\mathrm{m}}[k], u_{\mathrm{s,max}}[k], i_{\mathrm{s,max}}[k]\right)^{\top} + \boldsymbol{n}_{\mathrm{x}}$$

and corresponding output vector

$$\boldsymbol{y}[k] := N_{\mathrm{y}} \left(i^d_{\mathrm{s,ref}}[k], i^q_{\mathrm{s,ref}}[k]\right)^{\top} + \boldsymbol{n}_{\mathrm{y}}$$

where N_{x}, N_{y}, and $\boldsymbol{n}_{\mathrm{x}}$, $\boldsymbol{n}_{\mathrm{y}}$ are the normalization matrices and vectors, respectively. The normalization parameters are chosen to normalize input and output values of $\boldsymbol{x}$ and $\boldsymbol{y}$ within the $[-1, 1]$ range, enhancing data processing, training efficacy, and mitigating overfitting by preventing dominance of large input values over smaller ones [90, Sec. 11.5.3].

A dataset consisting of 2000000 sample sets was generated using MATLAB's `fmincon` function and parallel toolbox with 50 MATLAB workers (meaning 50 parallel MATLAB instances) on a multi-CPU workstation with 64 cores. This dataset generation process took approximately five days. Various input sample sets are created using Latin Hypercube (LH) sampling [91]. The inputs' ranges are detailed in Table 2. These input sample sets are subsequently utilized during the solution process of the NLP (13) that is based on the nonlinear machine model [recall (8) to (10)] and LUT-based representations of the machine nonlinearities, such as flux linkages (see Fig. 2), iron resistance or machine torque. Note that the current values used do not exceed the rated current since the flux maps are not available for currents beyond the rated current limit. This precaution is taken to prevent the NLP solver from failing (especially at high currents).

TABLE 2. Key Parameters of MO-HPO Framework

Data set generation	
Training data set size	1 500 000 samples
Validation data set size	500 000 samples
Input range $m_{\mathrm{m,ref}}$	-25.465…25.465 Nm
Input range ω_{m}	0…209.439 rad/s
Input range $u_{\mathrm{s,max}}$	186.161…310.269 V
Input range $i_{\mathrm{s,max}}$	6…9 A
Sampling method	latin hypercube sampling
ANN training	
Batch size b	64
Maximum validation failure runs	6
Optimizer	Adam
Initial learning rate l_{r}	$1 \cdot 10^{-3}$
MO-HPO	
Numbers m of hidden layers	1…5
Numbers n_i of neurons per hidden layer	
1 hidden layer	1…200
2…5 hidden layers	1…100
Optimization constraints	
N_{flops}	≤ 10000
MSE e_{y}	≤ 0.5
Optimization algorithm	NSGA-II
Population size	80
Number of generations	10

B. ANN ARCHITECTURE AND TRAINING

Fig. 5 illustrates a generic ANN architecture comprising input vector $\boldsymbol{x}$ and output vector $\widehat{\boldsymbol{y}} = (\widehat{i}^d_{\mathrm{s,ref}}, \widehat{i}^q_{\mathrm{s,ref}})^\top$ for the OFTC problem [recall (11) and Fig. 3]. This ANN architecture consists of one input layer, m hidden layers, and one output layer. The number of neurons in the ith hidden layer is denoted by n_i with $i \in \{0, \ldots, m+1\}$, where 0 and $m+1$ represent the input and output layer, respectively. The numbers of neurons in the input and output layers correspond to n_0 (representing the number of inputs $\boldsymbol{x}$) and n_{m+1} (representing the number of outputs $\widehat{\boldsymbol{y}}$), respectively. Thus, the ANN architecture is characterized by the number m of hidden layers and the numbers $n_1, n_2, \ldots, n_m$ of neurons in each hidden layer. The output of the ith neuron in the jth layer is given by

$$\widehat{y}_{j,i} = \Phi_{j,i}\left(\boldsymbol{w}_{j,i}^\top \boldsymbol{x}_j + b_{j,i}\right) \in \mathbb{R} \tag{14}$$

with input vector $\boldsymbol{x}_j := (x_{j,1}, \ldots, x_{j,n_{j-1}})^\top$ of the jth layer, weights vector $\boldsymbol{w}_{j,i} := (w^1_{j,i}, \ldots, w^{n_{j-1}}_{j,i})^\top$, bias $b_{j,i}$ and activation function $\Phi_{j,i}$. An artificial neuron with output as in (14) is illustrated in Fig. 4.

Collecting all n_j neurons' outputs of the jth layer in one output vector of the jth layer yields

$$\widehat{\boldsymbol{y}}_j = \boldsymbol{\Phi}_j\left(\boldsymbol{W}_j \boldsymbol{x}_j + \boldsymbol{b}_j\right) \in \mathbb{R}^{n_j} \tag{15}$$

with activation function, bias vector, and weighting matrix

$$\boldsymbol{\Phi}_j := \begin{pmatrix} \Phi_{j,1} \\ \vdots \\ \Phi_{j,n_j} \end{pmatrix}, \; \boldsymbol{b}_j := \begin{pmatrix} b_{j,1} \\ \vdots \\ b_{j,n_j} \end{pmatrix}, \; \boldsymbol{W}_j := \begin{bmatrix} \boldsymbol{w}_{j,1}^\top \\ \vdots \\ \boldsymbol{w}_{j,n_j}^\top \end{bmatrix}. \tag{16}$$

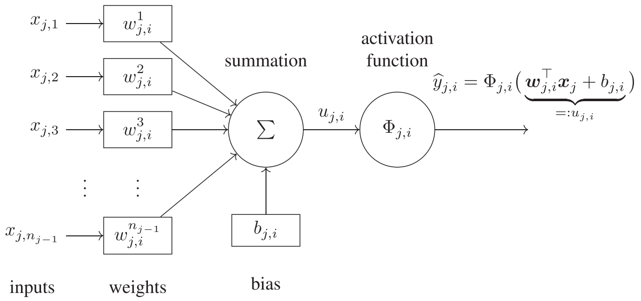


FIGURE 4. Illustration of *i*th artificial neuron in the *j*th layer.

Considering the output vector of the previous layer as input vector of the actual layer, i.e., $\boldsymbol{x}_j = \boldsymbol{y}_{j-1}$, the output of the ANN can be calculated recursively by

$$\left.\begin{array}{c} \hat{y} = \Phi_{m+1}(W_{m+1}x_{m+1} + b_{m+1}) = f_{w,b,\Phi}(x) \\ \overbrace{\Phi_m(W_m x_m + b_m)} \\ \vdots \\ \overbrace{\Phi_1(W_1 x_1 + b_1)} \\ \overbrace{\Phi_0(W_0 x_0 + b_0)} \end{array}\right\} \tag{17}$$

More detailed derivations can be found in [2]. For OFTC in particular and electrical drive applications in general, where low computational burdens are crucial, simple activation functions are essential. Therefore, the identity function for input and output layers, and the rectifying linear unit (ReLU) for the hidden layers are selected (for details, see [2]).

For this study, the selected training method is the *Adaptive Moment Estimation (Adam)* algorithm, which is based on stochastic gradient descent (SGD), and combines the advantages of SGD with the momentum method and adaptive learning rate adjustments [83]. An initial learning rate of $l_{\mathrm{r}} = 1 \times 10^{-3}$ is chosen. Given the substantial size of the generated dataset, batched training is employed, enabling the utilization of smaller subsets, referred to as batches having batch size b, for each update step in the ANN training process. In Table 2, all ANN training parameters are summarized.

C. MULTIOBJECTIVE ANN-HYPERPARAMETER OPTIMIZATION

Our goal is to find the nondominated Pareto solutions of the following MO-HPO problem described by the function:

$$\boldsymbol{f} : \mathbb{D}^u \times \mathbb{U} \to \mathbb{T}^2 \tag{18}$$

with

$$\mathbb{U} = \{1, 2, 3, 4, 5\}, \quad \mathbb{D} = \begin{cases} [1, 200], & \text{for } u = 1 \\ [1, 100]^u, & \text{for } u > 1 \end{cases}$$

where the use case space $\mathbb{U}$ comprises the five investigated use cases $u \in \mathbb{U}$ representing the depth of the ANN by its number m of hidden layers, i.e., $u = m$ (recall Fig. 5). The design space $\mathbb{D}$ comprises the use case dependent admissible number n_i with $i \in \{1, \ldots, m\}$ of neurons per layer ranging

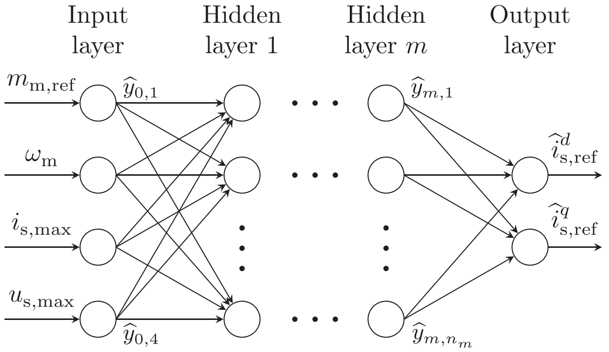


FIGURE 5. **Illustration of the ANN architectures with four input layer neurons, two output layer neurons and *m* hidden layers with their numbers of neurons per hidden layer $(n_1, n_2, \ldots, n_m)^\top \in \mathbb{N}^m$.**

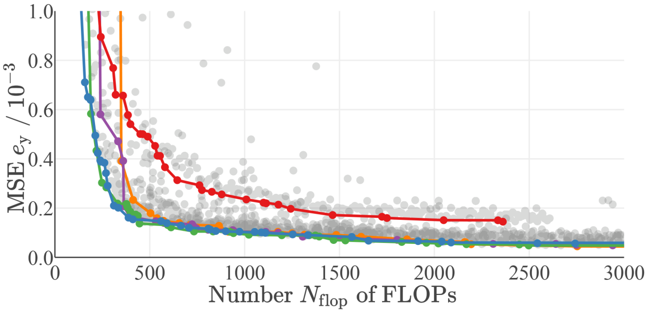


FIGURE 6. **Target space of MO-HP O results with dominated solutions [●] and Pareto optimal ANN designs with 1 hidden layers [—●—], 2 hidden layers [—●—], 3 hidden layers [—●—], 4 hidden layers [—●—] and 5 hidden layers [—●—].**

from 0 to 200 (if $u = m = 1$) or from 0 to 100 (if $u = m > 1$). The considered objectives (targets) are the *number of flops* N_{flop} and the *MSE* e_{y}, leading to the target space $\mathbb{T} \subseteq \mathbb{R}^2$ with the target vector $\boldsymbol{t} = (N_{\mathrm{flop}}, e_{\mathrm{y}})^\top$.

The MO-HPO problem

$$\begin{aligned} \boldsymbol{p}_{\mathrm{opt}} &:= \arg\min_{\boldsymbol{p}} \boldsymbol{f}(\boldsymbol{p}) \quad \forall\, \boldsymbol{p} \in \mathbb{D} \\ \text{s.t.} \quad & N_{\mathrm{flop}} \leq 10000 \\ & e_{\mathrm{y}} \leq 0.5 \end{aligned} \tag{19}$$

is solved individually for each use case (which means for each number of hidden layers), where the design vector $\boldsymbol{p} = (n_1, \ldots, n_u)^\top \in \mathbb{D}^u$ describes the numbers of neurons n_i in each hidden layer $i \in \{1, \ldots, u\}$ of the ANN. The constraints are selected heuristically, to condense the target space, and to keep MSE and number of FLOPs within realistic and application-specific boundaries.

The previously mentioned long training time, the cardinality and dimension of the design space, and the complexity of the MO-HPO problem motivate the use of an efficient and effective numerical search algorithm to identify the Pareto optimal ANN architectures with respect to several contradictory target variables. A well-established algorithm used for MO-HPO is the nondominated sorting genetic algorithm II (NSGA-II) [80], [86]. The MO-HPO in this study was performed based on the NSGA-II implementation of the Python package *pymoo* [92]. In the proposed framework, individual MO-HPO runs are executed for each use case separately allowing a later-on analysis of Pareto front solutions as a function of ANN depth. An empirical hyperparameter study resulted in the NSGA-II configuration with a population size of 80 and a number of generations of 10. Population size defines the number of samples evaluated in parallel per generation. The first generation is initialized based on Latin Hyper Cube (LHC) sampling, whereas later generations utilize results of the prior generation to identify the set of nondominated solutions, i.e. the Pareto front. Further details of the algorithm are given in [86]. The MO-HPO parameters are summarized in Table 2. On the used a AMD Ryzen 7 5800x CPU with associated NVIDIA RTX A5000 GPU, solving the MO-HPO problem took roughly 9 days in total.

The MO-HPO results are presented in Fig. 6. It shows results of the investigated ANN designs in the target space (i.e., MSE e_y over number N_{flop} of FLOPs) with dominated individual solutions [●] and Pareto optimal (nondominated) solutions per use case u (highlighted in different colors): $u = 1$ hidden layer [—●—], $u = 2$ hidden layers [—●—], $u = 3$ hidden layers [—●—], $u = 4$ hidden layers [—●—], $u = 5$ hidden layers [—●—]. With an increasing number of FLOPs the MSE decreases for all use cases. For $N_{\mathrm{flop}} > 800$, the MSE of the Pareto optimal designs of the use cases $u = 2, 3, 4$ and 5 converge to very small values, whereas, for the use case $u = 1$, the MSE converges to a significantly larger value.

D. VALIDATION

The applicability and transferability of the proposed approach is validated in two steps. First, representative ANN designs are selected (as a subset from Fig. 6) in both dominated and nondominated regimes. Then, measurements and results for these designs are presented to compare experimental values with model-based predictions.

1) MO DECISION MAKING

For experimental validation, four ANN designs are chosen to be implemented and tested on a realtime system in the laboratory. Fig. 7 shows the MO-HPO results of use case $u = 2$ (2 hidden layers) and the selected ANN designs: three Pareto optimal designs 4-4-4-2 [◆], 4-4-14-2 [◆], 4-75-12-2 [◆], and a non-Pareto optimal design 4-2-97-2 [◆].

2) EXPERIMENTAL RESULTS

The above selected ANN designs are trained and then implemented on a realtime dSpace system (see Fig. 8). In a first step, details for one selected ANN design are presented on a representative basis. In a second step, results of all (in equivalent manner performed) measurements are compared with their accordingly obtained theoretical predictions to conclude the validation.

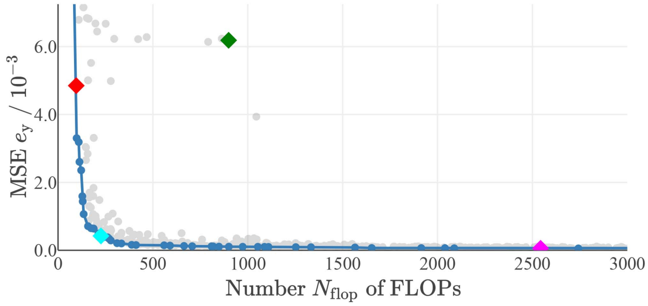


FIGURE 7. **MO-HPO results of two hidden layers with the selected ANN designs: 4-4-4-2 [◆], 4-2-97-2 [◆], 4-4-14-2 [◆], 4-75-12-2 [◆].**

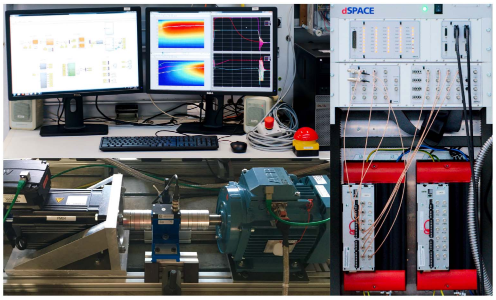


FIGURE 8. **Experimental setup.**

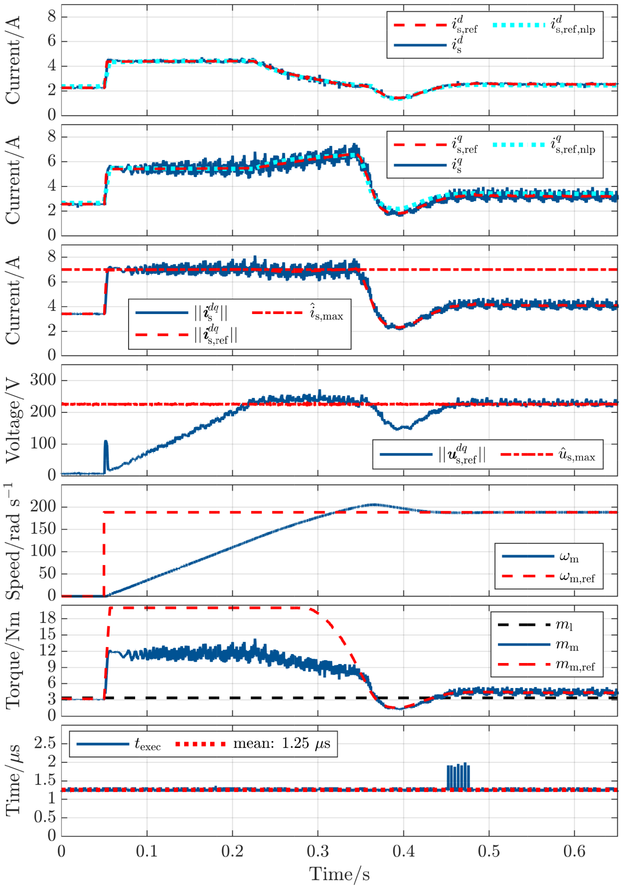


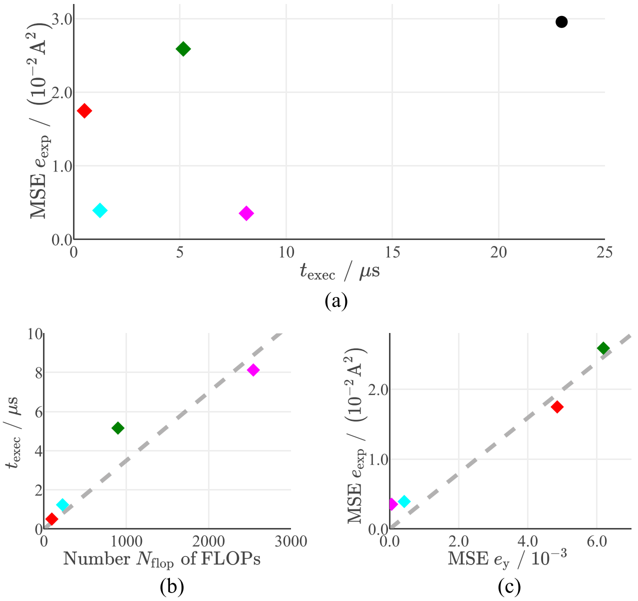
FIGURE 9. **Experimental results for the proposed ANN-based OFTC (architecture 4-4-14-2) for a nonlinear RSM (from top to bottom): Time series of actual currents $\boldsymbol{i}_{\rm s}$, ANN-based OFTC reference currents $\boldsymbol{i}_{\rm s,ref}$ and NLP-based OFTC reference currents $\boldsymbol{i}_{\rm s,ref,nlp}$ (*d* and *q*), current magnitudes of actual $||i^{dq}_{\rm s}||$ and reference currents $||i^{dq}_{\rm s,ref}||$, voltage magnitude $||u^{dq}_{\rm s}||$, angular velocity $\omega_{\rm m}$, torque $m_{\rm m}$, and execution time $t_{\rm exec}$.**

a) Scenario and Time Series: The implementation of the ANN is realized by a (self-written) MATLAB function, which is subsequently integrated into Simulink for the eventual real-time code generation. The 1.9 kW nonlinear RSM (right machine in Fig. 8) is speed controlled. The speed controller outputs the reference torque to the ANN-based OFTC system which computes the optimal reference currents for the underlying current controllers. For those, a generic and nonlinear current controller structure based on I/O-linearization was applied as introduced in [47, Sec. 6.6.7] (in German) and adapted in [93] for RSMs. Please note that any other current control system could be adopted as well. The prime mover is current (torque) controlled and acts as a load. Industrial inverters (see Fig. 8 bottom right) with switching frequency $f_{\rm sf} = 8$kHz are used to apply the reference voltages from the current controllers to the machines.

The considered experimental scenario (shown in Fig. 9 for ANN architecture 4-4-14-2) represents a typical start-up operation of an electrical drive system. Beginning from a standstill, a step change in the reference speed is applied, targeting a speed of $\omega_{\rm m,ref} = 125\%\, \omega_{\rm m,N}$ (125% of rated speed). Throughout the entire experiment, the load torque applied by the prime mover is kept constant at ≈ 3Nm.

Fig. 9 shows the experimental results of the ANN-based OFTC approach if the Pareto optimal design 4-4-14-2 [◆] (recall Fig. 7). Signals shown (with color code) are actual values [——], reference values [- - -] and maximum values [-·-·] of stator currents $i^d_{\rm s}$ and $i^q_{\rm s}$, current magnitude $\|\boldsymbol{i}^{dq}_{\rm s}\|$, voltage magnitude $\|\boldsymbol{u}^{dq}_{\rm s}\|$, machine speed $\omega_{\rm m}$, torque $m_{\rm m}$, and execution time $t_{\rm exec}$. The reference currents $i^d_{\rm s,ref,nlp}$ and $i^q_{\rm s,ref,nlp}$ [····] are also illustrated to show the accuracy of the ANN-based OFTC compared to the optimal solution of the NLP problem (13) obtained by MATLAB's `fmincon`. Note that $i^d_{\rm s,ref,nlp}$ and $i^q_{\rm s,ref,nlp}$ were added during postprocessing as `fmincon` cannot be used in realtime.

From $t = 0.0$ s to $t = 0.05s$ the machine is at stand-still while reference and produced torque satisfy the load torque, i.e., $m_{\rm m,ref} = m_{\rm m} = m_{\rm l}$. Moreover, no voltage or current constraints are violated. At $t = 0.05s$, a reference speed step to $\omega_{\rm m,ref} = 125\%\, \omega_{\rm m,N}$ is applied. Consequently, the reference torque jumps to the maximally feasible torque of ≈ 12 Nm. Note that the reference torque is larger than the maximally feasible torque. From $t = 0.05s$ to $t = 0.22s$, the requested maximally feasible torque is produced while the current limit is ensured by the ANN-based OFTC. At $t = 0.22s$, the voltage limit is reached. Therefore, from $t = 0.22s$ to $t = 0.36s$, the maximally feasible torque is reduced. At $t = 0.45s$, the reference speed is reached and the reference torque reduces



(a)

(b)

(c)

FIGURE 10. Experimental results of (a) MSE e_{exp} and execution time t_{exec} and the relation between MO-HPO and experimental results by (b) comparison of the execution time and number of FLOPs, (c) the MSEs e_{exp} and e_{y} of the selected ANN designs: 4-4-4-2 [◆], 4-2-97-2 [◆], 4-4-14-2 [◆], 4-75-12-2 [◆] and ORCC [•].

to the load torque mL and some additional friction torque. Consequently, the current magnitude reduces and the current limit is not touched anymore. From $t = 0.45s$ to $t = 0.65s$, voltage limit and reference torque are both satisfied. Besides, the current limit is also not reached. Throughout the experiment, the execution time of the ANN does not exceed 2 μs. The average execution time is solely 1.25 μs.

b) Comparison of all Selected ANNs and Analytical ORCC: The experiment is repeated for the four selected ANN designs (recall Fig. 7) and for the state-of-the-art optimal reference current computation (ORCC) [48], which computes the optimal reference currents analytically and in realtime.

For all selected ANNs and the analytical ORCC, the MSE

$$e_{\text{exp}} = \frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{i}^{dq}_{\text{s,ref}}[k] - \boldsymbol{i}^{dq}_{\text{s,ref,nlp}}[k]\|^2 \tag{20}$$

with the total number of samples K which are recorded during the experiment are computed and plotted against the mean execution time t_{exec} in Fig. 10(a). The ANN architecture with 4-75-12-2 [◆] has the lowest error e_{exp} but yields the highest computation time of the ANNs. In contrast, architecture 4-4-4-2 [◆] is the fastest but with higher error. Architecture 4-4-14-2 [◆] strikes a balance between computation time t_{exec} and MSE e_{exp}. For ANN architecture 4-2-97-2 [◆], it is possible to find ANNs with either lower error or lower computation time (dominated solution).

Fig. 10(b) shows the relationship between the total number N_{flop} of FLOPs and execution time t_{exec}. The computation time clearly increases with the number of FLOPs. In Fig. 10(c), the MSEs e_{exp} and e_{y} of experiment and MO-HPO are compared revealing a correlation between their error magnitudes. Finally, Fig. 10(a) clearly indicates that ORCC, with an average computation time of $t_{\text{exec}} \approx 22.96\,\mu s$ and a MSE of $e_{\text{exp}} \approx 2.96 \cdot 10^{-2}\text{A}^2$, significantly performs worse than all ANN-based OFTC approaches.

IV. DISCUSSION

A novel methodology for systematic Pareto-optimal ANN hyperparameter identification was effectively applied to OFTC for nonlinear SM. It is shown that ANN-based OFTC solutions can be obtained that achieve the best possible trade-off between execution time and accuracy as competing goals. Such solutions meet the reference torque demand or at least the maximum feasible torque while satisfying voltage and current constraints and accounting for machine nonlinearities (such as magnetic saturation and cross-coupling).

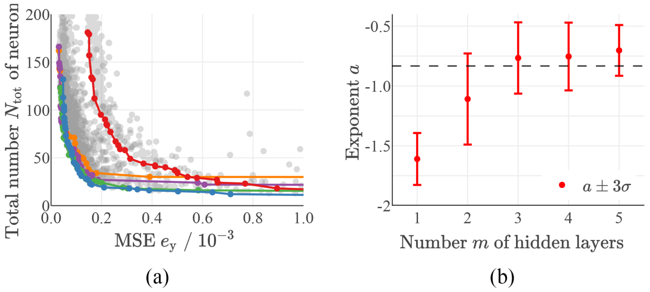
The study confirms the target conflict between an achievable MSE e_{y} and the number N_{flop} of FLOPs as a measure of required computational power. It also demonstrates the benefits of systematic ANN MO-HPO as an equally effective and efficient way to identify and implement OFTC solutions.

Applying results from the theory of approximation [71], allows to gain further insights into specific design rules for ANN-based OFTC approaches: In function fitting, three error types occur in general, i.e. training error (arising from non-convex optimization), estimation error (due to finite dataset size and generalization), and approximation error (related to expressivity capability of the chosen approach, in our case ANNs). Based on our ANN training approach with small initial learning rates and a sufficiently large dataset with two million samples, expressivity is dominant. From established works [71], [94], in this context, universal properties can be demonstrated. Specifically, the required total number of neurons N_{tot} within an ANN to approximate a given unknown (in our case OFTC defined) function to a defined accuracy (here the *expressivity error* e_{y}) asymptotically follows a power law:

$$N_{\text{tot}} \geq c \cdot \left(e_{\text{y}}\right)^{a} \quad \Longleftrightarrow \quad e_{\text{y}} \leq \sqrt[a]{\frac{N_{\text{tot}}}{c}}. \tag{21}$$

This allows to identify the scaling factor c and the exponent a by linear regression on the log-log transformation $\log(N_{\text{tot}}) \geq \log(c) + a \cdot \log(e_{\text{y}})$. In (21) represents one of the fundamental ANN approximation theory results. It not only specifies the asymptotic power law relationship between the lower bound for the total number N_{tot} of required neurons for a desired MSE, but also states the scaling factor c and exponent a to differ for shallow ($u = 1$) and deep ($u > 1$) ANNs. From the approximation theory approach, it follows that these parameters for deep ANNs are independent of the network depth u. In our case, Pareto fronts represent the best possible trade-offs and thereby should represent the expected lower boundaries.

In Fig. 11(a), the MSE e_{y} and the total number N_{tot} of neurons of all hidden layers are shown. For $N_{\text{tot}} \gg 100$, the MSE of the Pareto optimal designs approach a MSE limit that is no longer dominated by expressivity rather than training



(a) (b)

FIGURE 11. Power law fitting for each number of hidden layers *m*: (a) Pareto fronts N_{tot} versus e_{y} and (b) exponent $a \pm 3\sigma$. The dashed line indicates their mean for $m \geq 2$.

TABLE 3. Identified Power Law Parameters *a* and *c* (Recall (21)) With Confidence ±3σ and *F*-Statistics With Corresponding *p*-Value for Pareto Optimal ANNs Per Number *m* of Hidden Layers

m	c	a	F	p
1	0.000 ± 0.000	-1.610 ± 0.217	0.001	0.98
2	0.002 ± 0.006	-1.109 ± 0.381	0.004	0.95
3	0.048 ± 0.139	-0.766 ± 0.298	0.014	0.91
4	0.054 ± 0.151	-0.754 ± 0.283	0.014	0.91
5	0.102 ± 0.211	-0.703 ± 0.212	0.013	0.91

residuals. For each number $m \in \{1, \ldots, 5\}$ of hidden layers, the power law was fitted to its Pareto front separately using the `curve_fit` implementation in Scipy [95]. The resulting parameters are summarized in Table 3 with uncertainties estimated according to [96]. To test the hypothesis of the power law dependence, an analysis of variance (ANOVA) was performed using `f_oneway` in [95]. Even if all data points (which are largely, but not completely, dominated by expressivity) of the Pareto fronts are taken into account, the predicted power law behavior of the (21) already shows a significance level of better than 10% for each of them. These ANNs may be interpreted to represent the lower bound of totally required neurons N_{tot} to achieve a defined MSE. Likewise, the expected difference in the exponent for shallow versus deep ANNs is significantly dissimilar, whereby the convergence behavior for deep ANNs can be classified as the same and thereby confirms the approximation theoretical prediction.

In Fig. 11(b), the determined exponents a, along with their estimator uncertainties (3σ) are shown. The figure illustrates that deep neural networks with $m \geq 2$ exhibit a common exponent. The mean value $\frac{1}{4}\sum_{m=2}^{5} a_m = -0.83$ [- - -] falls within the uncertainty interval of $m \geq 2$. In contrast, shallow networks with only one hidden layer ($m = 1$) result in a significantly different exponent.

In practical application, this implies that 1) utilizing deep neural networks (with a number of hidden layers greater than 1, i.e., $m \geq 2$) can significantly reduce the MSE e_{y} at the same number of neurons N_{tot}, compared to shallow networks (with only one hidden layer, i.e. $m = 1$). Additionally, 2) it is sufficient to consider only ANNs with two hidden layers $m = 2$ as a surrogate for this application, because with the same total number of neurons no further MSE improvement can be expected for any deeper ANN.

V. CONCLUSION

A novel workflow for the design of ANN-based solutions tailored to the unique demands of electrical drives has been proposed, implemented and validated. The workflow encompasses supervised learning of ANNs and the structured determination of optimal ANN architectures through a multiobjective optimization approach. This framework was successfully applied to OFTC of nonlinear RSMs.

The obtained ANN-based OFTC designs have demonstrated superior accuracy and execution time when compared to state-of-the-art analytical computational methods. Moreover, invoking insights from approximation theory allowed to provide valuable design rules for optimal ANN architectures, which significantly reduce the design space and enhance the efficiency of the design process of the ANN-based OFTC approach for electrical drives.

Future work will extend ANN-based OFTC by including temperature variations for a more comprehensive analysis of electrical drives. In addition, the framework's applicability will be broadened by also covering EESM with additional inputs and outputs. Finally, to make this workflow more accessible, we are working on an open-source Python toolchain facilitating its implementation and promoting collaborative innovations within the research community.

REFERENCES

[1] S. Zhang, O. Wallscheid, and M. Porrmann, "Machine learning for the control and monitoring of electric machine drives: Advances and trends," *IEEE Open J. Ind. Appl.*, vol. 4, pp. 188–214, Jun. 2023, doi: 10.1109/ojia.2023.3284717.

[2] M. A. Buettner, N. Monzen, and C. M. Hackl, "Artificial neural network based optimal feedforward torque control of interior permanent magnet synchronous machines: A feasibility study and comparison with the state-of-the-art," *Energies*, vol. 15, no. 5, 2022, Art. no. 1838, doi: 10.3390/en15051838.

[3] M. Wishart and R. Harley, "Identification and control of induction machines using artificial neural networks," in *Proc. Conf. Rec. IEEE Ind. Appl. Conf. 28th IAS Annu. Meeting*, 1993, pp. 703–709, doi: 10.1109/ias.1993.298875.

[4] Y. Kung, C. Liaw, and M. Ouyang, "Adaptive speed control for induction motor drives using neural networks," *IEEE Trans. Ind. Electron.*, vol. 42, no. 1, pp. 25–32, Feb. 1995, doi: 10.1109/41.345842.

[5] S. Weerasooriya and M. El-Sharkawi, "Identification and control of a DC motor using back-propagation neural networks," *IEEE Trans. Energy Convers.*, vol. 6, no. 4, pp. 663–669, Dec. 1991, doi: 10.1109/60.103639.

[6] M. A. Rahman and M. A. Hoque, "Online self-tuning ANN-based speed control of a PM DC motor," *IEEE/ASME Trans. Mechatron.*, vol. 2, no. 3, pp. 169–178, Sep. 1997, doi: 10.1109/3516.622969.

[7] G. Book et al., "Transferring online reinforcement learning for electric motor control from simulation to real-world experiments," *IEEE Open J. Power Electron.*, vol. 2, pp. 187–201, Mar. 2021, doi: 10.1109/ojpel.2021.3065877.

[8] T.-J. Ren and T.-C. Chen, "Robust speed-controlled induction motor drive based on recurrent neural network," *Electric Power Syst. Res.*, vol. 76, no. 12, pp. 1064–1074, 2006, doi: 10.1016/j.epsr.2006.01.002.

[9] A. Rubaai and M. Kankam, “Adaptive tracking controller for induction motor drives using online training of neural networks,” *IEEE Trans. Ind. Appl.*, vol. 36, no. 5, pp. 1285–1294, Sep./Oct. 2000, doi: 10.1109/28.871276.

[10] A. Rubaai, R. Kotaru, and M. Kankam, “Online training of parallel neural network estimators for control of induction motors,” *IEEE Trans. Ind. Appl.*, vol. 37, no. 5, pp. 1512–1521, Sep./Oct. 2001, doi: 10.1109/28.952529.

[11] X. Fu and S. Li, “A novel neural network vector control technique for induction motor drive,” *IEEE Trans. Energy Convers.*, vol. 30, no. 4, pp. 1428–1437, Dec. 2015, doi: 10.1109/tec.2015.2436914.

[12] A. Ba-Razzouk, A. Cheriti, G. Olivier, and P. Sicard, “Field oriented control of induction motors using neural networks decouplers,” in *Proc. IECON 95-21st Annu. Conf. IEEE Ind. Electron.*, 1995, pp. 1428–1433, doi: 10.1109/iecon.1995.484160.

[13] M. Rahman and M. Hoque, “On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors,” *IEEE Trans. Energy Convers.*, vol. 13, no. 4, pp. 311–318, Dec. 1998, doi: 10.1109/60.736315.

[14] Y. Yi, D. M. Vilathgamuwa, and M. A. Rahman, “Implementation of an artificial-neural-network-based real-time adaptive controller for an interior permanent-magnet motor drive,” *IEEE Trans. Ind. Appl.*, vol. 39, no. 1, pp. 96–104, Jan./Feb. 2003, doi: 10.1109/tia.2002.807233.

[15] B. Burton, R. Harley, G. Diana, and J. Rodgerson, “Implementation of a neural network to adaptively identify and control VSI-fed induction motor stator currents,” *IEEE Trans. Ind. Appl.*, vol. 34, no. 3, pp. 580–588, May/Jun. 1998, doi: 10.1109/28.673729.

[16] S. Li, H. Won, X. Fu, M. Fairbank, D. C. Wunsch, and E. Alonso, “Neural-network vector controller for permanent-magnet synchronous motor drives: Simulated and hardware-validated results,” *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3218–3230, Jul. 2020, doi: 10.1109/tcyb.2019.2897653.

[17] M. G. Simoes and B. K. Bose, “Neural network based estimation of feedback signals for a vector controlled induction motor drive,” in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 1994, pp. 471–479, doi: 10.1109/ias.1994.345442.

[18] V. Tipsuwanporn, C. Tarasantisuk, A. Numsumran, and W. Sawaengsinkasikit, “Motor speed identification using multilayer feedforward neural networks,” in *Proc. IEEE 4th Int. Conf. Power Electron. Drive Syst. - Indonesia Proc.*, 2001, pp. 62–65, doi: 10.1109/peds.2001.975285.

[19] H. Tsai, A. Keyhani, J. A. Demcko, and D. A. Selin, “Development of a neural network based saturation model for synchronous generator analysis,” *IEEE Trans. Energy Convers.*, vol. 10, no. 4, pp. 617–624, Dec. 1995, doi: 10.1109/60.475831.

[20] R. M. Tallam, T. G. Habetler, and R. G. Harley, “Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics,” *IEEE Trans. Power Electron.*, vol. 17, no. 6, pp. 1089–1095, Nov. 2002, doi: 10.1109/tpel.2002.805611.

[21] X. Huang, T. Gabetler, and R. Harley, “Detection of rotor eccentricity faults in closed-loop drive-connected induction motors using an artificial neural network,” in *Proc. IEEE 35th Annu. Power Electron. Specialists Conf.*, 2004, pp. 913–918, doi: 10.1109/pesc.2004.1355541.

[22] S. Mohagheghi, R. Harley, T. Habetler, and D. Divan, “Condition monitoring of power electronic circuits using artificial neural networks,” *IEEE Trans. Power Electron.*, vol. 24, no. 10, pp. 2363–2367, Oct. 2009, doi: 10.1109/tpel.2009.2017806.

[23] L. Guo and L. Parsa, “Model reference adaptive control of five-phase IPM motors based on neural network,” in *Proc. IEEE Int. Electric Machines Drives Conf.*, 2011, pp. 563–568, doi: 10.1109/iemdc.2011.5994871.

[24] M. Dybkowski and K. Klimkowski, “Artificial neural network application for current sensors fault detection in the vector controlled induction motor drive,” *Sensors*, vol. 19, no. 3, pp. 1–16, 2019, doi: 10.3390/s19030571.

[25] K. Jankowska and M. Dybkowski, “Design and analysis of current sensor fault detection mechanisms for PMSM drives based on neural networks,” *Designs*, vol. 6, no. 1, pp. 1–20, 2022, doi: 10.3390/designs6010018.

[26] M. Wlas, Z. Krzeminski, J. Guzinski, H. Abu-Rub, and H. A. Toliyat, “Artificial-neural-network-based sensorless nonlinear control of induction motors,” *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 520–528, Sep. 2005. [Online]. Available: https://ieeexplore.ieee.org/document/1495523/

[27] M. Mohamadian, E. Nowicki, F. Ashrafzadeh, A. Chu, R. Sachdeva, and E. Evanik, “A novel neural network controller and its efficient DSP implementation for vector-controlled induction motor drives,” *IEEE Trans. Ind. Appl.*, vol. 39, no. 6, pp. 1622–1629, Nov./Dec. 2003, doi: 10.1109/tia.2003.819441.

[28] N. Monzen and C. M. Hackl, “Artificial neural network based optimal feedforward torque control of electrically excited synchronous machines,” in *Proc. IEEE Int. Symp. Ind. Electron.*, 2023, pp. 1–8.

[29] I. Boldea, L. Tutelea, Lucian N. Parsa, and D. Dorrell, “Automotive electric propulsion systems with reduced or no permanent magnets: An overview,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5696–5711, Oct. 2014, doi: 10.1109/tie.2014.2301754.

[30] E. Schmidt, “Synchronous reluctance machines with high-anisotropy rotors–comparison of their operational characteristics,” in *Proc. Australas. Universities Power Eng. Conf.*, 2014, pp. 1–6, doi: 10.1109/AUPEC.2014.6966496.

[31] P. Niazi, H. A. Toliyat, and A. Goodarzi, “Robust maximum torque per ampere (MTPA) control of PM-assisted SynRM for traction applications,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 1538–1545, Jul. 2007, doi: 10.1109/TVT.2007.896974.

[32] S.-Y. Jung, J. Hong, and K. Nam, “Current minimizing torque control of the IPMSM using Ferrari’s method,” *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 5603–5617, Dec. 2013, doi: 10.1109/TPEL.2013.2245920.

[33] M. Preindl and S. Bolognani, “Optimal state reference computation with constrained MTPA criterion for PM motor drives,” *IEEE Trans. Power Electron.*, vol. 30, no. 8, pp. 4524–4535, Aug. 2015, doi: 10.1109/TPEL.2014.2354299.

[34] J. Lemmens, P. Vanassche, and J. Driesen, “PMSM drive current and voltage limiting as a constraint optimal control problem,” *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 3, no. 2, pp. 326–338, Jun. 2015, doi: 10.1109/JESTPE.2014.2321111.

[35] G. Schoonhoven and M. N. Uddin, “MTPA- and FW-based robust nonlinear speed control of IPMSM drive using Lyapunov stability criterion,” *IEEE Trans. Ind. Appl.*, vol. 52, no. 5, pp. 4365–4374, Sep./Oct. 2016, doi: 10.1109/TIA.2016.2564941.

[36] A. Dianov, F. Tinazzi, S. Calligaro, and S. Bolognani, “Review and classification of MTPA control algorithms for synchronous motors,” *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 3990–4007, Apr. 2022, doi: 10.1109/TPEL.2021.3123062.

[37] J. Ahn et al., “Field weakening control of synchronous reluctance motor for electric power steering,” *IET Electric Power Appl.*, vol. 1, no. 4, pp. 565–570, 2007, doi: 10.1049/iet-epa:20060212.

[38] B. Cheng and T. Tesch, “Torque feedforward control technique for permanent-magnet synchronous motors,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 969–974, Mar. 2010, doi: 10.1109/TIE.2009.2038951.

[39] M. Tursini, E. Chiricozzi, and R. Petrella, “Feedforward flux-weakening control of surface-mounted permanent-magnet synchronous motors accounting for resistive voltage drop,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 440–448, Jan. 2010, doi: 10.1109/TIE.2009.2034281.

[40] M. Preindl and S. Bolognani, “Model predictive direct torque control with finite control set for PMSM drive systems, part 2: Field weakening operation,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 648–657, May 2013, doi: 10.1109/TII.2012.2220353.

[41] J. Kim, I. Jeong, K. Nam, J. Yang, and T. Hwang, “Sensorless control of PMSM in a high-speed region considering iron loss,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6151–6159, Oct. 2015, doi: 10.1109/TIE.2015.2432104.

[42] P. Zhang, D. M. Ionel, and N. A. O. Demerdash, “Saliency ratio and power factor of IPM motors with distributed windings optimally designed for high efficiency and low-cost applications,” *IEEE Trans. Ind. Appl.*, vol. 52, no. 6, pp. 4730–4739, Nov. 2016, doi: 10.1109/TIA.2016.2598313.

[43] N. Urasaki, T. Senjyu, and K. Uezato, “A novel calculation method for iron loss resistance suitable in modeling permanent-magnet synchronous motors,” *IEEE Trans. Energy Convers.*, vol. 18, no. 1, pp. 41–47, Mar. 2003, doi: 10.1109/TEC.2002.808329.

[44] C. Cavallaro, A. O. D. Tommaso, R. Miceli, A. Raciti, G. R. Galluzzo, and M. Trapanese, “Efficiency enhancement of permanent-magnet synchronous motor drives by online loss minimization approaches,” *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1153–1160, Aug. 2005, doi: 10.1109/TIE.2005.851595.

[45] R. Ni, D. Xu, G. Wang, L. Ding, G. Zhang, and L. Qu, "Maximum efficiency per ampere control of permanent-magnet synchronous machines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2135–2143, Apr. 2015, doi: 10.1109/TIE.2014.2354238.

[46] H. Eldeeb, C. M. Hackl, L. Horlbeck, and J. Kullick, "A unified theory for optimal feedforward torque control of anisotropic synchronous machines," *Int. J. Control*, vol. 91, pp. 2273–2302, 2017, doi: 10.1080/00207179.2017.1338359.

[47] C. Hackl, J. Kullick, and N. Monzen, "Optimale betriebsführung für nichtlineare synchronmaschinen," in *Elektrische Antriebe*, R. von J. A. Böcker and R. G. Griepentrog, Eds. Berlin, Germany: Springer-Verlag, 2020, pp. 1006–1068.

[48] C. M. Hackl, J. Kullick, and N. Monzen, "Generic loss minimization for nonlinear synchronous machines by analytical computation of optimal reference currents considering copper and iron losses," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2021, pp. 1348–1355, doi: 10.1109/icit46573.2021.9453497.

[49] O. Haala, B. Wagner, M. Hofmann, and M. Marz, "Optimal current control of externally excited synchronous machines in automotive traction drive applications," *Int. J. Elect., Electron. Commun. Sci.*, vol. 7, pp. 1133–1139, 2013, doi: 10.5281/ZENODO.1335628.

[50] O. Korolova, P. Duck, A. Brune, J. Jurgens, and B. Ponick, "Prediction of efficiency-optimized salient-pole synchronous machines operating range using a coupled numerical-analytical method," in *Proc. Int. Conf. Elect. Mach.*, 2014, pp. 871–876, doi: 10.1109/icelmach.2014.6960283.

[51] S. Muller and N. Parspour, "Applying a mea;surement-based iron loss model to an efficiency optimized torque control of an electrically excited synchronous machine," in *Proc. Int. Conf. Elect. Mach.*, 2020, pp. 779–785, doi: 10.1109/icem49940.2020.9271029.

[52] C. Zaghrini, G. Khoury, M. Fadel, R. Ghosn, and F. Khatounian, "Minimum copper losses per torque optimization on electrically excited synchronous motors for electric vehicles applications," in *Proc. IEEE 20th Int. Power Electron. Motion Control Conf.*, 2022, pp. 661–666, doi: 10.1109/pemc51159.2022.9962853.

[53] R. Grune, "Verlustoptimaler betrieb einer elektrisch erregten synchronmaschine für den einsatz in elektrofahrzeugen," PhD thesis, Technischen Universität Berlin, 2012.

[54] A. Brune, "Elektrisch erregte synchronmaschinen als fahrantrieb," PhD thesis, Leibniz Univeritält Hannover, 2018.

[55] J. Tang and Y. Liu, "Comparison of copper loss minimization and field current minimization for electrically excited synchronous motor in mild hybrid drives," in *Proc. 19th Eur. Conf. Power Electron. Appl.*, 2017, pp. P.1–P.10, doi: 10.23919/epe17ecceeurope.2017.8099352.

[56] D. Uzel, K. Zeman, Z. Peroutka, and M. Danek, "Optimal vector control for wound rotor salient pole synchronous motor up to base speed," in *Proc. 15th Int. Power Electron. Motion Control Conf.*, 2012, pp. DS2a.12a–1-DS2a, doi: 10.1109/epepemc.2012.6397263.

[57] G. Marques and M. F. Iacchetti, "Minimum loss conditions in a salient-pole wound-field synchronous machine drive," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc.*, 2019, pp. 1183–1189, doi: 10.1109/iecon.2019.8926793.

[58] Y. Kim and K. Nam, "Copper-loss-minimizing field current control scheme for wound synchronous machines," *IEEE Trans. Power Electron.*, vol. 32, no. 2, pp. 1335–1345, Feb. 2017, doi: 10.1109/tpel.2016.2547953.

[59] X. Wang, Q. Wen, J. Wu, J. Yang, X. Zhao, and Z. Wang, "A novel neural network and sensitivity analysis method for predicting the thermal resistance of heat pipes with nanofluids," *Appl. Thermal Eng.*, vol. 236, Jan. 2024, Art. no. 121677, doi: 10.1016/j.applthermaleng.2023.121677.

[60] F. Mohammadi, Z. Yavari, M. R. Nikoo, A. Al-Nuaimi, and H. Karimi, "Machine learning model optimization for removal of steroid hormones from wastewater," *Chemosphere*, vol. 343, Dec. 2023, Art. no. 140209, doi: 10.1016/j.chemosphere.2023.140209.

[61] K. Naderi, A. Foroughi, and A. Ghaemi, "Analysis of hydraulic performance in a structured packing column for air/water system: RSM and ANN modeling," *Chem. Eng. Process. - Process Intensification*, vol. 193, Nov. 2023, Art. no.109521, doi: 10.1016/j.cep.2023.109521.

[62] S. B. Akdaş and A. Fişne, "A data-driven approach for the prediction of coal seam gas content using machine learning techniques," *Appl. Energy*, vol. 347, Oct. 2023, Art. no. 121499, doi: 10.1016/j.apenergy.2023.121499.

[63] K. Pijackova et al., "Genetic algorithm designed for optimization of neural network architectures for intracranial EEG recordings analysis," *J. Neural Eng.*, vol. 20, no. 3, Jun. 2023, Art. no. 036034, doi: 10.1088/1741-2552/acdc54.

[64] S. M. Tabatabaei, N. Attari, S. A. Panahi, M. Asadian-Pakfar, and B. Sedaee, "EOR screening using optimized artificial neural network by sparrow search algorithm," *Geoenergy Sci. Eng.*, vol. 229, Oct. 2023, Art. no. 212023, doi: 10.1016/j.geoen.2023.212023.

[65] J. Suto, "The effect of hyperparameter search on artificial neural network in human activity recognition," *Open Comput. Sci.*, vol. 11, no. 1, pp. 411–422, Jun. 2021, doi: 10.1515/comp-2020-0227.

[66] K. Rahman, S. Gopalakrishnan, B. Fahimi, A. V. Rajarathnam, and M. Ehsani, "Optimized torque control of switched reluctance motor at all operational regimes using neural network," *IEEE Trans. Ind. Appl.*, vol. 37, no. 3, pp. 904–913, May/Jun. 2001, doi: 10.1109/28.924774.

[67] N. Bulic, E. Krasser, and I. Erceg, "Neural network based excitation control of synchronous generator," in *Proc. EUROCON Int. Conf. Comput. Tool*, 2007, pp. 1935–1941, doi: 10.1109/eurcon.2007.4400352.

[68] F. F. M. El-Sousy and K. A. Abuhasel, "Nonlinear robust optimal control via adaptive dynamic programming of permanent-magnet linear synchronous motor drive for uncertain two-axis motion control system," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1940–1952, Mar./Apr. 2020, doi: 10.1109/tia.2019.2961637.

[69] J. Yu, P. Shi, W. Dong, B. Chen, and C. Lin, "Neural network-based adaptive dynamic surface control for permanent magnet synchronous motors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 640–645, Mar. 2015, doi: 10.1109/tnnls.2014.2316289.

[70] M. Schenke, W. Kirchgassner, and O. Wallscheid, "Controller design for electrical drives by deep reinforcement learning: A proof of concept," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4650–4658, Jul. 2020, doi: 10.1109/tii.2019.2948387.

[71] I. Gühring, M. Raslan, and G. Kutyniok, "Expressivity of deep neural networks," 2020, *arXiv:2007.04759*.

[72] G. Bramerdorfer, J. A. Tapia, J. J. Pyrhonen, and A. Cavagnino, "Modern electrical machine design optimization: Techniques, trends, and best practices," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 7672–7684, Oct. 2018, doi: 10.1109/TIE.2018.2801805.

[73] G. Lei, J. Zhu, Y. Guo, C. Liu, and B. Ma, "A review of design optimization methods for electrical machines," *Energies*, vol. 10, no. 12, 2017, Art. no. 1962, doi: 10.3390/en10121962.

[74] T. Orosz et al., "Robust design optimization and emerging technologies for electrical machines: Challenges and open problems," *Appl. Sci.*, no. 19, vol. 10, 2020, Art. no. 6653, doi: 10.3390/app10196653.

[75] Z. Xia, Z. Liu, and J. M. Guerrero, "Multi-objective optimal model predictive control for three-level ANPC grid-connected inverter," *IEEE Access*, vol. 8, pp. 59590–59598, 2020, doi: 10.1109/access.2020.2981996.

[76] T. Liu et al., "Multi-objective model predictive control of grid-connected three-level inverter based on hierarchical optimization," *Chin. J. Elect. Eng.*, vol. 7, no. 1, pp. 63–72, 2021, doi: 10.23919/cjee.2021.000006.

[77] A. Bakeer, G. Magdy, A. Chub, and D. Vinnikov, "Predictive control based on ranking multi-objective optimization approach for quasi z-source inverter," *CSEE J. Power Energy Syst.*, vol. 7, no. 6, pp. 1152–1160, Nov. 2021, doi: 10.17775/cseejpes.2020.01310.

[78] M. G. Villarreal-Cervantes, A. Rodriguez-Molina, C.-V. Garcia-Mendoza, O. Penaloza-Mejia, and G. Sepulveda-Cervantes, "Multi-objective on-line optization approach for the DC motor controller tuning using differential evolution," *IEEE Access*, vol. 5, pp. 20393–20407, 2017, doi: 10.1109/access.2017.2757959.

[79] M. Moallem, B. Mirzaeian, O. Mohammed, and C. Lucas, "Multi-objective genetic-fuzzy optimal design of PI controller in the indirect field oriented control of an induction motor," *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3608–3612, Sep. 2001, doi: 10.1109/20.952673.

[80] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8043–8093, Aug. 2023, doi: 10.1007/s10462-022-10359-2.

[81] M. Parsa, J. P. Mitchell, C. D. Schuman, R. M. Patton, T. E. Potok, and K. Roy, "Bayesian multi-objective hyperparameter optimization for accurate, fast, and efficient neural network accelerator design," *Front. Neurosci.*, vol. 14, pp. 1–18, Jul. 2020, doi: 10.3389/fnins.2020.00667.

[82] D. Schröder and M. Buss, *Intelligente Verfahren : Identifikation Und Regelung Nichtlinearer Systeme*. Berlin, Germany: Springer-Verlag, 2017.

[83] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[84] L. Prechelt, *Early Stopping - But When?*, Berlin, Heidelberg: Springer, 1998. doi: 10.1007/3-540-49430-8_3.

[85] N. Palm, M. Landerer, and H. Palm, “Gaussian process regression based multi-objective bayesian optimization for power system design,” *Sustainability*, vol. 14, no. 19, 2022, Art. no. 12777, doi: 10.3390/su141912777.

[86] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[87] J. Müller, “Socemo: Surrogate optimization of computationally expensive multiobjective problems,” *INFORMS J. Comput.*, vol. 29, no. 4, pp. 581–596, 2017, doi: 10.1287/ijoc.2017.0749.

[88] H. Palm and L. Arndt, “Reinforcement learning-based hybrid multi-objective optimization algorithm design,” *Information*, vol. 14, no. 5, pp. 1–13, 2023, doi: 10.3390/info14050299.

[89] C. M. Hackl, *Non-Identifier Based Adaptive Control in Mechatronics: Theory and Application (ser. Lecture Notes in Control and Information Sciences)*, no. 466. Berlin: Springer International Publishing, 2017. [Online]. Available: https://www.springer.com/de/book/9783319550343

[90] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning (ser. Springer Series in Statistics)*. New York, NY, USA: Springer, 2009. [Online]. Available: https://link.springer.com/10.1007/978-0-387-84858-7

[91] R. L. Iman, J. C. Helton, and J. E. Campbell, “An approach to sensitivity analysis of computer models: Part I—Introduction, input variable selection and preliminary variable assessment,” *J. Qual. Technol.*, vol. 13, no. 3, pp. 174–183, 1981, doi: 10.1080/00224065.1981.11978748.

[92] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89497–89509, 2020, doi: 10.1109/ACCESS.2020.2990567.

[93] S.-W. Su, H. Börngen, C. M. Hackl, and R. Kennel, “Nonlinear current control of reluctance synchronous machines with analytical flux linkage prototype functions,” *IEEE Open J. Ind. Electron. Soc.*, vol. 3, pp. 582–593, Sep. 2022, doi: 10.1109/OJIES.2022.3208329.

[94] A. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993, doi: 10.1109/18.256500.

[95] P. Virtanen et al., “SciPy 1.0: Fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.

[96] K. W. Vugrin, L. P. Swiler, R. M. Roberts, N. J. Stucky-Mack, and S. P. Sullivan, “Confidence region estimation techniques for nonlinear regression in groundwater flow: Three case studies,” *Water Resour. Res.*, vol. 43, no. 3, Mar. 2007, Art. no. 2005WR004804, doi: 10.1029/2005WR004804.

NIKLAS MONZEN received the B.Eng. degree in electrical engineering from Ostfalia University of Applied Sciences, Wolfenbüttel/Braunschweig, Germany, in 2014, the M.Sc. degree in electrical engineering from the Technical University of Munich (TUM), Munich, Germany, in 2017. Since 2020, he has been a Ph.D. candidate.

His expertise lies in nonlinear modeling and optimal control of electrical drives, primarily focusing on various synchronous machines, including reluctance synchronous machines, permanent magnet synchronous machines, and electrically excited synchronous machines. As a member of the Research Institute for Sustainable Energy Systems (ISES) and the Laboratory for Mechatronic and Renewable Energy Systems (LMRES), the University of Applied Sciences in Munich, he conducts research under Prof. Christoph Hackl’s guidance.

FLORIAN STROEBL received the B.Eng. degree in electrical engineering and the M.Eng. degree in electrical and microsystems engineering from the University of Applied Sciences (UAS) OTH Regensburg, Germany, in 2019 and 2020, respectively.

In 2021, he began his doctoral studies as a Research Assistant with the Institute for Sustainable Energy Systems (ISES), Munich, Germany, as part of a publicly funded research program. His special interest include the optimization of highly cost intensive and time consuming experiments with application focus on lithium ion battery cell degradation for aging analysis.

HERBERT PALM received the Ph.D. degree in physics from the Friedrich-Alexander-Universität (FAU), Erlangen-Nürnberg, Germany, in 1994.

He worked in leading international R&D and operations positions of both high-tech start-ups and corporate companies until 2008. In 2008, he took over a full-time professorship with the University of Applied Sciences Munich (UAS Munich). Since 2014, he has been a head of the UAS Munich Systems Engineering Master’s program. His research activities primarily focus on analysis, design, and optimization of complex systems. He calls VUCA environment his home. From an application point of view, he focuses on the support of projects with a sustainable energy system link. Prof. Palm is one of the founding members of the Institute of Sustainable Energy Systems (ISES) in Munich. He is a senior member of INCOSE and the German Physical Society (DPG).

CHRISTOPH M. HACKL (Senior Member, IEEE) was born in 1977 in Mannheim, Germany. After studying Electrical Engineering (with focus on mechatronics and systems and control) from Technical University of Munich (TUM), Munich, Germany, and the University of Wisconsin-Madison, Wisconsin, USA, he received the B.Sc., Dipl.-Ing., and Dr.-Ing. (Ph.D.) degrees in electrical engineering in 2003, 2004, and 2012, respectively, from TUM, Munich, Germany.

Since 2004, he has been teaching electrical drives, power electronics, and mechatronic & renewable energy systems. Since 2014, he has been the head of the research group “Control of Renewable Energy Systems (CRES)” at TUM. In 2018, he became a Professor for Electrical Machines and Drives and the head of the “Laboratory for Mechatronic and Renewable Energy Systems (LMRES)” with the Hochschule München (HM) University of Applied Sciences, Munich, Germany. In 2019, he completed his habilitation on “Mechatronic and Renewable Energy Systems” at TUM and co-founded the research Institute for Sustainable Energy Systems (ISES) at HM, which he co-heads since then. His research interests include nonlinear, adaptive and optimal control and design of electrical drives, and mechatronic and renewable energy systems.