Artificial neural network based optimal feedforward torque control of electrically excited synchronous machines

Niklas Monzen \(^1\) and Christoph Hackl \(^2\)

\(^1\)HM Munich University of Applied Sciences
\(^2\)Affiliation not available

October 30, 2023

Abstract

An Artificial Neural Network (ANN) based Optimal Feedforward Torque Control (OFTC) strategy for electrically excited synchronous machines (EESMs) is proposed. After design, data set creation, training and validation of the ANN, the analytical computation of the optimal stator and exciter currents is achieved which allows to minimize copper and iron losses and to produce the desired (or maximally feasible) machine torque. Voltage and currents constraints of stator and exciter are considered as well. In contrast to conventional OFTC, the proposed ANN-based OFTC strategy does not require iterations nor a decision tree to find the optimal current triple while machine nonlinearities, magnetic cross-coupling, saturation and speed-dependent iron losses are taken into account. In addition, the proposed ANN design procedure allows to consider measurable OFTC goals and computational resources that ensures a real-time capable implementation. Comprehensive simulation results for a real and nonlinear EESM clearly show these benefits by comparing the proposed ANN-based OFTC with results of a nonlinear optimization problem (NLP) solver (which cannot be used in real-time).
Artificial neural network based optimal feedforward torque control of electrically excited synchronous machines

1st Niklas Monzen and 2nd Christoph M. Hackl

Laboratory for Mechatronic and Renewable Energy Systems (LMRES)
HM Munich University of Applied Sciences
Munich, Germany
{niklas.monzen, christoph.hackl}@hm.edu

Abstract—An Artificial Neural Network (ANN) based Optimal Feedforward Torque Control (OFTC) strategy for electrically excited synchronous machines (EESMs) is proposed. After design, data set creation, training and validation of the ANN, the analytical computation of the optimal stator and exciter currents is achieved which allows to minimize copper and iron losses and to produce the desired (or maximally feasible) machine torque. Voltage and currents constraints of stator and exciter are considered as well. In contrast to conventional OFTC, the proposed ANN-based OFTC strategy does not require iterations nor a decision tree to find the optimal current triple while machine nonlinearities, magnetic cross-coupling, saturation and speed-dependent iron losses are taken into account. In addition, the proposed ANN design procedure allows to consider measurable OFTC goals and computational resources that ensures a real-time capable implementation. Comprehensive simulation results for a real and nonlinear EESM clearly show these benefits by comparing the proposed ANN-based OFTC with results of a nonlinear optimization problem (NLP) solver (which cannot be used in real-time).

Index Terms—Artificial neural network (ANN), copper & iron losses, efficiency, electrically excited synchronous machine (EESM), machine learning, nonlinear machine model, optimal feedforward torque control (OFTC), optimal reference current computation, optimization.

I. INTRODUCTION

Electrical machines are widely used in a variety of applications such as industry, mobility and transportation or robotics. Optimal feedforward torque control (OFTC) is crucial for almost all applications as it assures optimal operation for all feasible operating points and increases the efficiency of the electrical drive systems [1, 2]. The main idea of OFTC is to provide optimal reference currents to the underlying current controllers. The computed optimal reference currents will assure that the reference (or at least the maximally feasible) torque is produced while losses are minimized and physical constraints (such as voltage and current limits) are satisfied.

This paper proposes a novel ANN-based OFTC for electrically excited synchronous machines (EESM).

OFTC of synchronous machine (SM), in particular of permanent magnet synchronous machine (PMSM) and reluctance synchronous machine (RSM), is currently a hot research topic and a lot of research has been done (see e.g., [3–27]). In [23], a unified theory was presented that covers different operating strategies such as Maximum-Torque-per-Ampere (MTPA; or Maximum-Torque-per-Current (MTPC)), Field Weakening (FW), Maximum Current (MC) and Maximum-Torque-per-Voltage (MTPV) and allows to compute the optimal stator reference currents for RSM and PMSM analytically without simplifying assumptions on flux linkage nonlinearities or stator resistances. However, iron losses were neglected. This has been overcome by the extended analytical approach presented in [17, 24]. But EESMs, with additional exciter current, are not covered.

In [28–30] numerical identification and search algorithms are proposed as offline OFTC approaches for EESM that are not suitable for a real-time implementation.

In [31], the MTPA problem for stator and exciter currents is solved by the use of a Lagrangian. However, electromagnetic cross-coupling and iron losses are not considered. Furthermore, the reference currents are computed offline and stored in look-up tables (LUTs) due to the computationally expensive solution of the MTPA problem.

In [32–36] online OFTC approaches are presented that are based on the linear machine model with constant inductances, i.e nonlinear saturation effects are neglected. In addition, iron losses are neglected. Only [36] considers iron losses but assuming them to be linear.

In [34–38], different MTPA approaches (e.g., maximum torque per copper losses, maximum torque per exciter current) are presented for the computation of optimal currents of EESMs but voltage and current limits are not taken into account.

Only [32, 33] present OFTC approaches that also consider voltage and current limits. However, in both are iron losses neglected and constant inductances are assumed. Furthermore, in [32] the stator resistance is neglected in the stator voltage constraint equation.

In this paper, we propose to use artificial neural networks (ANNs) for the implementation of an ANN-based OFTC of EESM such that the computational load is reduced and the real-time capability is assured.
In the field of electrical drives, there are numerous ANN-based approaches for different applications (see detailed literature reviews in [39, 40]). Works related to their application to EESM, are rarely found. Only in [41], the active power damping of a synchronous generator is realized by an ANN that generates the exciter reference voltage. An ANN-based OFTC for a highly nonlinear anisotropic (I)PMSM is presented in [40]. However, a detailed topology study on how to find an optimal topology with respect to real-time capability, computation time and approximation errors, is still missing. Furthermore, the presented approach is not suitable for EESM because (at least) two additional inputs (exciter current and voltage constraints) and one outputs (reference exciter current) are required for ANN-based OFTC of EESMs.

Hence, to the best knowledge of the authors, this is the very first paper that proposes ANN-based OFTC of EESMs that considers (i) arbitrary nonlinearities (e.g., saturation, \((d, q)\)-cross-coupling and cross-coupling between stator and exciter), (ii) iron and copper losses, (iii) stator and exciter voltage and current limits (iii) computational resources for a real-time capable implementation.

II. MODELING AND PROBLEM STATEMENT

A. Nonlinear electrically excited synchronous machine model

The nonlinear machine model of the EESM in rotating \((d, q)\)-reference frame is shown in Fig. 1. The flux linkages are transformer-like coupled by the stator and rotor iron core. Therefore, the stator flux linkages \(\psi_s^{dq} = \psi_s^{d} (i_s^{d,e} , \omega_m)\) and exciter flux linkage \(\psi_e = \psi_e (i_s^{d,e} , \omega_m)\) depend on the mechanical angular velocity \(\omega_m\) and on all currents \(i_s^{d,e}\) where \(i_s^{d}, i_s^{q}\) are the stator \((d, q)\) currents and \(i_e\) is the exciter current. Neglecting flux leakage effects so that

with \(J := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\), \(\Omega_2 = (0, 0)^T\), stator voltages \(u_s^{dq} := (u_s^{d}, u_s^{q})^T\), stator currents \(i_s^{dq} := (i_s^{d}, i_s^{q})^T\), stator resistance matrix \(R_s^{dq} \in \mathbb{R}^{2 \times 2}\), stator iron currents \(i_s^{dq} = (i_s^{d,e}, i_s^{d,e})^T\), exciter voltage \(u_e\), exciter current \(i_e\), exciter resistance \(R_e\) and mechanical angular velocity \(\omega_m = \frac{\omega}{n_p}\) (i.e., electrical angular velocity \(\omega\), divided by pole pair number \(n_p\)). In addition, the introduced current and frequency dependent stator iron resistance matrix \(R_{s,Fe}^{dq} = R_{s,Fe}^{d} (i_s^{d,e}, \omega_p) \in \mathbb{R}^{2 \times 2}\) can be found by

\[
R_{s,Fe}^{dq} = \frac{2}{3\kappa} \left( \begin{array}{cc} \omega_p J \psi_s^{d} + \frac{d}{dt} \psi_s^{dq} \end{array} \right)^T \left( \begin{array}{cc} \omega_p J \psi_s^{d} + \frac{d}{dt} \psi_s^{dq} \end{array} \right) I_2 \tag{2}
\]

with \(I_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}\), the nonlinear iron losses \(p_{Fe}(i_s^{d,e}, \omega_m)\), and the Clarke transformation factor \(\kappa \in \{2/3, \sqrt{2/3}\}\). Combining (1) and the conservation of energy & power condition yields

\[
 \begin{align*}
 p_{Cu} &= \frac{2}{3\kappa} (i_s^{dq})^T R_{s,Fe}^{d} i_s^{dq} + R_e i_s^{dq}^2, \\
 p_{Fe} &= \frac{2}{3\kappa} (i_s^{dq})^T R_{s,Fe}^{d} i_s^{dq}, \\
 m_{m} &= \frac{2\kappa}{3\kappa} (i_s^{dq} + i_s^{dq})^T J \psi_s^{dq}, \\
 p_L &= p_{Cu} + p_{Fe}
 \end{align*}
 \tag{3}
\]

with the stator and exciter copper losses \(p_{Cu}\), the overall nonlinear machine losses \(p_L(i_s^{d,e})\) and the nonlinear machine torque \(m_{m}(i_s^{d,e}, \omega_m)\).

B. Problem statement

OFTC generates the optimal reference currents

\[
\begin{align*}
 i_{s,ref}^{d,e}(\bullet) &= \begin{bmatrix} i_s^{d,ref} (m_{m,ref}, \omega_m, u_{s,max}, i_s^{max}, u_e^{max}, i_e^{max}) \\
 i_s^{q,ref} (m_{m,ref}, \omega_m, u_{s,max}, i_s^{max}, u_e^{max}, i_e^{max}) \\
 i_e^{ref} (m_{m,ref}, \omega_m, u_{s,max}, i_s^{max}, u_e^{max}, i_e^{max}) \\
 \end{bmatrix},
 \end{align*}
 \tag{4}
\]

where \(i_s^{ref}, i_e^{ref}\) are the stator \((d, q)\)-reference currents and \(i_e^{ref}\) is the exciter reference current. The reference currents are functions of reference torque \(m_{m,ref}\), mechanical angular velocity \(\omega_m = \frac{\omega}{n_p}\), stator constraints \(u_{s,max}\) and \(i_s^{max}\) (maximum voltage and current magnitude, respectively) and exciter constraints \(u_e^{max}\) and \(i_e^{max}\) (maximum voltage and current, respectively), see Fig. 2.

The main goals of OFTC are (i) satisfaction of all machine constraints, (ii) generation of the reference (or, at least, the maximally feasible) machine torque, and (iii) minimization of (machine) losses.

The machine constraints are satisfied if

\[
\begin{align*}
 \|u_s^{dq}(i_s^{d,e})\| &\leq u_{s,max}, \quad \|i_s^{dq}\| \leq i_{s,max}, \\
 |u_e^{dq}| &\leq u_{e,max}, \quad |i_e| \leq i_{e,max}
 \end{align*}
 \tag{5}
\]

hold. The optimal (feasible) torque is achieved by minimizing the deviation \(|m_{m,ref} - m_{m}|\) between reference torque \(m_{m,ref}\) and actual machine torque \(m_{m}\). An efficient (loss minimal) operation is ensured by minimizing all (machine) losses \(p_L := p_{Cu} + p_{Fe}\) including copper \(p_{Cu}\) and iron \(p_{Fe}\) losses, respectively.
III. ANN-BASED OFTC of EESMs

To solve the OFTC problem for EESMs, in this work, an artificial neural network will be trained and validated to obtain an ANN-based OFTC algorithm.

A. Dataset creation

The ANN is trained by supervised learning using training and validation data sets extracted from the overall data set

\[(Y; X) := \left((y[1]; x[1]), \ldots, (y[K]; x[K])\right), \tag{6}\]

which contains \(k \in \{1, \ldots, K\}\) sample sets

\[(y[k]; x[k])\]

of corresponding output \(y[k]\) and input \(x[k]\) vectors. For each sample input vector

\[x[k] := \left(m_{m, \text{ref}}[k], \omega_m[k], u_{s, \text{max}}[k], i_{s, \text{max}}[k], \right.\]
\[\left. u_{e, \text{max}}[k], i_{e, \text{max}}[k]\right)^\top, \tag{7}\]

the corresponding sample output vector

\[y[k] := \left(i_{s, \text{ref}}^{d}[k], i_{s, \text{ref}}^{q}[k], i_{e, \text{ref}}[k]\right)^\top\]

must be found by solving the nonlinear optimization problem (NLP; which cannot be solved in real-time!)

\[\hat{i}_{s,e,\text{ref}}^{d}[k] = \arg \min_{\hat{i}_{s,e}^{d}[k]} p_L(\hat{i}_{s,e}^{d}[k])\]
\[\text{s.t.} \quad \|u_{s}^{d}(\hat{i}_{s,e}^{d}[k])\| \leq u_{s, \text{max}}[k],\]
\[\|\hat{i}_{s,e}^{d}[k]\| \leq i_{s, \text{max}}[k],\]
\[|u_e[\hat{i}_{s,e}^{d}[k]]| \leq u_{e, \text{max}}[k],\]
\[|\hat{i}_e[k]| \leq i_{e, \text{max}}[k],\]
\[m_{m}^{d}(\hat{i}_{s,e}^{d}[k]) = \bar{m}_{m, \text{ref}}(\hat{i}_{s,e}^{d}[k])\]  \(\tag{8}\)

iteratively for all \(k \in \{1, \ldots, K\}\) samples. The NLP in (8) can be solved by e.g. MATLAB’s function \texttt{fmincon} and minimizes the machine losses while voltage and current limits in stator and exciter and limited reference torque \(\bar{m}_{m, \text{ref}}\) are satisfied. The “limited” but feasible reference torque

\[\bar{m}_{m, \text{ref}}[k] := \text{sat}_{m_{\text{max}}, \text{max}}(m_{m, \text{ref}}[k]). \tag{9}\]

is saturated to the maximally feasible torque \(m_{m, \text{max}}[k]\) (which depends on actual operating point, and voltage and currents limits; recall inequality constraints in (8)).

The equality constraint in (8) guarantees that the possible torque deviation \(|m_{m, \text{ref}}[k] - m_{m}[k]| \neq 0\) (between desired torque and produced torque) is minimized if the reference torque \(m_{m, \text{ref}}[k]\) is not feasible and only the limited but feasible reference torque \(\bar{m}_{m, \text{ref}}[k] \neq m_{m, \text{ref}}[k]\) can be applied. The maximally feasible torque \(m_{m, \text{max}}[k]\) is the solution to the sub-NLP (also solved by e.g. \texttt{fmincon}) given by

\[m_{m, \text{max}}[k] = \min_{\hat{i}_{s,e}^{d}[k]} \text{sign}(m_{m, \text{ref}}[k]) \|m_{m}^{d}(\hat{i}_{s,e}^{d}[k])\|\]
\[\text{s.t.} \quad \|u_{s}^{d}(\hat{i}_{s,e}^{d}[k])\| \leq u_{s, \text{max}}[k],\]
\[\|\hat{i}_{s,e}^{d}[k]\| \leq i_{s, \text{max}}[k],\]
\[|u_e[\hat{i}_{s,e}^{d}[k]]| \leq u_{e, \text{max}}[k],\]
\[|\hat{i}_e[k]| \leq i_{e, \text{max}}[k].\]  \(\tag{10}\)

Finally, each solution \(y[k] := \hat{i}_{s,e,\text{ref}}^{d}[k]\) of the overall NLP as in (8) with corresponding input \(x[k]\) as in (7) is collected as sample set \((y[k]; x[k])\) and appended to the dataset as in (6).

B. ANN architecture

Fig. 3 shows the architecture of a generic ANN with input \(x\) and output \(\hat{y} := \left(\hat{y}^{d}_{s, \text{ref}}, \hat{y}^{q}_{s, \text{ref}}, \hat{y}^{e}_{\text{ref}}\right)^\top\). The ANN consists of one input layer, \(m \in \mathbb{N}\) hidden layers and one output layer. The number of neurons of the \(i\)-th hidden layer is \(n_i \in \mathbb{N}\), where \(i \in \{0, 1, \ldots, m+1\}\) where 0 and \(m + 1\) are the indices of the input and output layer, respectively. The numbers of neurons per input and output layer matches the number \(n_0\) of inputs \(x\) and the number \(n_{m+1}\) of outputs \(\hat{y}\), respectively. Consequently, the ANN-architecture is defined by the number \(m\) of hidden layers and the numbers \(n_1, n_2, \ldots, n_m \in \mathbb{N}\) of neurons per hidden layer.

The output of the \(i\)-th neuron of the \(j\)-th layer is given by

\[\hat{y}_{j,i} = \Phi_{j,i}(w_{j,i}x_j + b_{j,i}) \in \mathbb{R}\]  \(\tag{11}\)
with input vector $x_j$ of the $j$-th layer, weights vector $w_{j,i}$, bias $b_{j,i}$, and activation function $\Phi_{j,i}$. The input layer is used for normalization such that the input values are normalized to the range between $-1$ and $1$; which improves data handling and training capability and reduces the risk of overfitting as it helps to prevent that large values of one input dominate smaller values of other values [42, Sec. 11.5.3]. As a result, for the outputs of the $i$-th input layer, (11) simplifies to

$$\hat{y}_{0,i} = \Phi_{0,i}(w_{0,i}x_i + b_{0,i}) \in \mathbb{R}$$  (12)

where $x_i$ is the $i$-th input. Due to the need for low computational burden in the considered application, simple activation functions such as the identity for input and output layers and the rectifying linear unit (ReLU) for the hidden layers are used, respectively (as illustrated in Fig. 4).

![Identity and Rectified Linear Unit](image)

**Figure 4:** Illustration of used activation functions & derivatives.

The output of the $j$-th layer is given by

$$\hat{y}_j = \begin{pmatrix} \hat{y}_{j,1} \\ \vdots \\ \hat{y}_{j,n_j} \end{pmatrix} = \begin{pmatrix} \Phi_{j,1}(w_{j,1}^\top x_j + b_{j,1}) \\ \vdots \\ \Phi_{j,n_j}(w_{j,n_j}^\top x_j + b_{j,n_j}) \end{pmatrix} \in \mathbb{R}^{n_j},$$  (13)

and the input vector $x_j$ of the $j$-th layer corresponds to the output vector $\hat{y}_{j-1}$ of the previous layer, i.e.

$$\forall \; j \in \{1, \ldots, m\} : \; x_j = \hat{y}_{j-1}. \tag{14}$$

Therefore, the output vector of the ANN is a function of all ANN parameters, i.e.

$$\hat{y} = \hat{y}_{m+1} = f(x, w_0, b_0, w, b)$$  (15)

with weights and biases collected in the following vectors

$$\begin{aligned} w_0 & := (w_{0,1}, \ldots, w_{0,n_0})^\top \in \mathbb{R}^{n_0}, \\
 b_0 & := (b_{0,1}, \ldots, b_{0,n_0})^\top \in \mathbb{R}^{n_0}, \\
 w & := (w_{1,1}, \ldots, w_{m,n,m+1})^\top \in \mathbb{R}^{\sum_{k=1}^{m+1} n_k - 1}, \\
 b & := (b_{1,1}, \ldots, b_{m,n,m+1})^\top \in \mathbb{R}^{\sum_{k=1}^{m+1} n_k}. \end{aligned}$$  (16)

As memory and storage are limited in real-time applications, the total number of parameters (all weights and biases)

$$N_{\text{param}} = \sum_{k=0}^{m+1} n_k + n_0 + \sum_{k=1}^{m+1} n_k - 1$$  (17)

must already be taken into account during the ANN design.

In addition, by evaluating (13)-(15) with respect to the mathematical operations required, the number of floating point operations (FLOPs) of the ANN can be computed as follows

$$N_{\text{flop}} = 2n_0 + 2n_{m+1}n_m + \sum_{k=1}^{m+1} n_k (2n_{k-1} + 1)$$  (18)

which allows to estimate the ANN’s execution time

$$t_{\text{exec}} = \frac{N_{\text{flop}}}{f_c}$$  (19)

with clock frequency $f_c$ & number $N_c$ of clock cycles per FLOP.

### C. ANN training

The training of each ANN architecture finds the best (optimal) parameters $(w^*, b^*)$ by minimizing the output’s mean squared error (MSE) $e_y$ of a dataset (recall (6)), i.e.

$$\begin{aligned} (w^*, b^*) &= \arg \min_{(w, b)} \frac{1}{K} \sum_{k=1}^{K} \| y[k] - \hat{y}[k] \|^2. \end{aligned}$$  (20)

There are different training algorithms (e.g., Gradient Descent [43, Sec. 10.4.4], Levenberg-Marquardt [43, Sec. 4.1]) that update the weights and biases iteratively. In this work, the Adaptive Moment Estimation (Adam) algorithm is used. It is based on Stochastic Gradient Descent (SGD) and combines the advantages of SGD with the momentum method and the adaptation of the learning rate [44].

To achieve proper training and to avoid overfitting of the ANN, the dataset is split into a training and validation set. The training set is used to train the ANN (adaption of its weights and biases), while the validation set is used to evaluate the approximation error of the ANN if facing unseen data. The training is stopped, when the validation error increases a certain number of consecutive evaluations (also known as maximum validation failure runs). The adapted ANN parameters, obtained before the validation error increases, are finally used as best weights and biases for ANN implementation.

### D. Performance evaluation of ANN by OFTC criteria

For the application-specific evaluation of the optimized ANN, the OFTC errors $e_{\text{OFTC}} := (e_{a_1}, e_{b_1}, e_{a_2}, e_{b_2}, e_m, e_L)^\top$
over a (test) dataset are computed as follows

\[
\begin{align*}
    e_u &= \frac{1}{K} \sum_{k=1}^{K} \max \left(0, \left| u \hat{d}_q[k] \right| - u_{\text{m,max}}[k] \right)^2, \\
    e_i &= \frac{1}{K} \sum_{k=1}^{K} \max \left(0, \left| i_{\text{dref}}[k] \right| - i_{\text{s,max}}[k] \right)^2, \\
    e_u' &= \frac{1}{K} \sum_{k=1}^{K} \max \left(0, \left| u_{\text{ref}}[k] \right| - u_{e,\text{max}}[k] \right)^2, \\
    e_i' &= \frac{1}{K} \sum_{k=1}^{K} \max \left(0, \left| i_{\text{ref}}[k] \right| - i_{e,\text{max}}[k] \right)^2, \\
    e_m &= \frac{1}{K} \sum_{k=1}^{K} \left( \hat{m}_m[k] - m_{\text{m,ref}}[k] \right)^2, \\
    e_L &= \frac{1}{K} \sum_{k=1}^{K} \left( \hat{p}_L[k] - p_L[k] \right)^2
\end{align*}
\]  

(21)

where \( u_{\hat{d}q} := u_d (i_{\text{s,ref}}), \) \( \hat{d}_q := (i_{\text{s,ref}}, i_{\text{s,ref}}) \), \( \hat{u}_e := u_e (i_{\text{ref}}, \hat{m}_m := m_m (i_{\text{s,ref}}) \) and \( \hat{p}_L := p_L (i_{\text{s,ref}}) \) represent approximated stator voltages, stator currents, exciter voltage, machine torque and losses obtained for the ANN output currents \( \hat{y} = i_{\text{s,ref}} \), respectively.

In terms of the main OFTC goals for EESMs (see Sec. II-B), the finding of an optimal ANN architecture is the trade-off between the three OFTC criteria: (i) minimizing each error of \( e_{\text{OFTC}} \), (ii) minimizing the number \( N_{\text{param}} \) of parameters (storage equivalent) and (iii) minimizing the number \( N_{\text{FLOP}} \) of FLOPs (execution time equivalent).

IV. VALIDATION

Finally, the proposed design procedure is applied. An ANN-based OFTC for a real nonlinear electrically excited synchronous machine is created. An optimal design is selected by a architecture study. The trained ANN is implemented and simulated in a closed-loop dynamic simulation.

A. ANN design

The investigated machine is a 10 kW EESM having the parameters listed in Tab. I. The nonlinear flux linkage maps were obtained by experimental measurements.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated mech. power</td>
<td>( p_{\text{m,R}} )</td>
<td>10 kW</td>
</tr>
<tr>
<td>Rated mech. speed</td>
<td>( \omega_{\text{m,R}} )</td>
<td>157 rad/s</td>
</tr>
<tr>
<td>Rated mech. torque</td>
<td>( m_{\text{R}} )</td>
<td>67 Nm</td>
</tr>
<tr>
<td>Maximum stator voltage</td>
<td>( u_{\text{s,max}} )</td>
<td>200 V</td>
</tr>
<tr>
<td>Maximum stator current</td>
<td>( i_{\text{s,max}} )</td>
<td>18 A</td>
</tr>
<tr>
<td>Maximum exciter voltage</td>
<td>( u_{e,\text{max}} )</td>
<td>100 V</td>
</tr>
<tr>
<td>Maximum exciter current</td>
<td>( i_{e,\text{max}} )</td>
<td>20 A</td>
</tr>
<tr>
<td>Stator resistance</td>
<td>( R_s )</td>
<td>0.38 ( \Omega )</td>
</tr>
<tr>
<td>Exciter resistance</td>
<td>( R_e )</td>
<td>0.83 ( \Omega )</td>
</tr>
</tbody>
</table>

The ANN is designed and trained for input ranges defined in Tab. II. MATLAB’s \texttt{fmincon} (MATLAB Version R2022b) is used for solving the NLP (8). A dataset with 573 750 samples is generated and separated into a training dataset (430 312 samples) and validation dataset (143 438 samples).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training dataset size</td>
<td>430 312 samples</td>
</tr>
<tr>
<td>Validation dataset size</td>
<td>143 438 samples</td>
</tr>
<tr>
<td>Input range ( m_{\text{m,ref}} )</td>
<td>0...84 Nm</td>
</tr>
<tr>
<td>Input range ( \omega_{\text{m}} )</td>
<td>0...212 rad/s</td>
</tr>
<tr>
<td>Input range ( u_{\text{s,ref}} )</td>
<td>139...218 V</td>
</tr>
<tr>
<td>Input range ( i_{\text{s,ref}} )</td>
<td>12...18 A</td>
</tr>
<tr>
<td>Input range ( i_{\text{e,ref}} )</td>
<td>16...20 A</td>
</tr>
</tbody>
</table>

Numbers of hidden layers: 1...4
Numbers of neurons per hidden layer: 2...200
FLOP limit: 1000...2000

Batch size: 50
Maximum number of epochs: 500
Maximum validation failure runs: 6
Optimizer: Adam
Initial learning rate: \( 1 \cdot 10^{-3} \) \footnote{Assuming a DSP (e.g. TMS320F28004x) with two clock cycles per FLOP and a clock frequency of \( f_s = 300 \) MHz gives an execution time of \( t_{\text{exec}} = 13 \) us, resulting in an utilization rate of 13\% for a controller sampling time of 10 kHz.}

For the considered EESM, the exciter voltage limit is never reached because \( R_e i_{\text{e,ref}} < u_{\text{s,ref}} \) holds for all operating conditions and, therefore, the ANN input \( u_{\text{s,ref}} \) is (and can be) neglected. To design a real-time capable ANN architecture, the maximum number of hidden layers, the maximum numbers of neurons per hidden layer and the maximum number of FLOPs are limited according Tab. II. As a result of this limitation, there are 1282 176 ANN architectures possible. Due to the limitation of training time, 200 ANN architectures are selected for training (50 architectures per number of hidden layers with the maximum number of FLOPs).

The training algorithm is implemented in Python using the PyTorch package with the training parameters specified in Tab. II. The trained 200 ANNs are evaluated according to the OFTC criteria given by MSES \( e_{\text{OFTC}} \) (recall (21)), the number of FLOPs \( N_{\text{FLOP}} \) (recall (18)) and the number of parameters \( N_{\text{param}} \) (recall (17)) in order to find eventually one optimal ANN architecture. The validation dataset is used for computing the MSES. Minimizing all these criteria at once guarantees to achieve the four main OFTC goals.

Fig. 5 shows the trained ANNs [●], the Pareto optimal ANNs [★] and the selected ANN [★] according to the OFTC criteria \( e_{u_e}, e_{i_e}, e_{u_s}, e_{i_s}, e_m, e_{L}, N_{\text{FLOP}} \) and \( N_{\text{param}} \).

The selected ANN has two hidden layers with \( n_1 = 9 \) and \( n_2 = 71 \) neurons per hidden layer. It is a Pareto optimal design and a trade-off between the OFTC criteria (i.e., the number of FLOPs \( N_{\text{FLOP}} = 1884 \), the number of parameters \( N_{\text{param}} = 990 \), the MSES \( e_{u_e} = 9.6 \cdot 10^{-5}, e_{i_e} = 2.1 \cdot 10^{-5}, e_{u_s} = 2.6 \cdot 10^{-5}, e_m = 2.8 \cdot 10^{-5}, e_{L} = 0.008 \cdot 10^{-5} \).
B. Closed-loop implementation and simulation

A closed-loop simulation with the selected ANN and the nonlinear machine model (1)-(3) of the investigated EESM is implemented in MATLAB & Simulink R2022b. An ideal pulse width modulation (PWM) inverter with switching frequency $f_{sw} = 8 \text{ kHz}$ is used to apply the reference voltages of the current controller to the machine. The ANN-based OFTC generates the reference currents for the current controllers. As an input for the OFTC, the reference torque is produced by the speed controller. The simulation scenario represents a start-up operation of an electrical drive system. After the machine is slowly ramped up from stand-still to the speed $\omega_m = 10\% \omega_{m,R}$, a reference speed step to $\omega_m = 100\% \omega_{m,R}$ is applied while the load torque $m_l = 20\% m_{m,R}$ is kept constant during the whole simulation scenario. The ANN-based OFTC results are compared to the results of MATLAB’s function fmincon that solves the NLP directly (which can’t be used in a real-time application).

Fig. 6 shows the time series plots of actual values (ANN-based [ - - - ] & NLP-based [ - - - - ]). reference values (ANN-based [ - - - ] & NLP-based [ - - - - - ]) and maximum values [ - - - - - - ] of stator currents $i_d$ and $i_q$, current magnitude $\|i_{dq}\|$, voltage magnitude $\|u_{dq}\|$, exciter current $i_e$, machine speed $\omega_m$ and torque $m_m$.

From $t = 0.25 \text{ s}$ to $t = 0.6 \text{ s}$ the machine is ramped up to $\omega_m = 10\% \omega_{m,R}$. The reference torque is satisfied while no voltage or current limits are touched. At $t = 0.81 \text{ s}$, a reference speed step to $\omega_m = \omega_{m,R}$ is applied. Consequently, the reference torque jumps up to the maximum torque. From $t = 0.81 \text{ s}$ to $t = 1.2 \text{ s}$, the requested maximum torque is produced while the stator current limit and exciter current limit are ensured by the ANN-based OFTC. At $t = 1.2 \text{ s}$, the stator voltage limit is reached. Therefore, from $t = 1.2 \text{ s}$ to $t = 2.0 \text{ s}$, the maximally feasible torque is reduced (the demanded reference torque can not be applied anymore). At $t = 2.2 \text{ s}$, the reference speed is reached and the reference torque reduces to $m_{m,ref} = m_l$. Consequently, the currents reduce and their limits are not touched anymore. From $t = 2.2 \text{ s}$ to $t = 2.5 \text{ s}$, the stator voltage limit is slowly ramped down (e.g., to emulate a decreasing battery voltage in electric vehicles). Nevertheless, the voltage limit and the reference torque can still be satisfied. Besides, the current limits are also not reached. During this simulation scenario, the ANN-based OFTC operates at all (physically possible) operation limits and satisfies the reference torque wherever it is feasible.

V. CONCLUSION

An artificial neural network (ANN)-based solution for the optimal feedforward torque control (OFTC) problem of nonlinear electrically excited synchronous machines (EESM) has been proposed. The approach allows to consider typical nonlinearities such as magnetic cross-coupling and saturation effects and iron losses. The ANN is trained and validated by a comprehensive training and validation data set consisting of 573750 solution points of the highly nonlinear OFTC optimization problem of EESMs (obtained by fmincon).

The proposed ANN-based OFTC of EESMs is implementable in real-time since during ANN design computation requirements (e.g., storage requirement, FLOPS and execution time) are already taken into account. In this regard, for a nonlinear EESM, several ANNs are optimized by a parameter study and analyzed with regard to measurable OFTC criteria.
Finally, an optimal ANN is selected from the obtained Pareto optimal designs and validated by realistic simulation. It is shown, that the ANN-based OFTC fulfils all OFTC goals: (i) voltage and current limits are satisfied, (ii) the reference torque is produced (if feasible according to constraints; otherwise the maximally feasible torque is produced) and (iii) iron and copper losses are minimized.

Future work will (i) validate the proposed methods by experiments, (ii) improve the ANN design by multi-objective optimization of ANN architecture and training, (iii) include further inputs (e.g., stator and exciter resistances) and (iii) study if ANNs are also capable of considering dynamic effects (such as current transients).

REFERENCES


