

Received June 8, 2021, accepted July 4, 2021, date of publication July 8, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095668

Artificial Neural Network and Data Dimensionality Reduction Based on Machine Learning Methods for PMSM Model Order Reduction

MARIA RALUCA RAIA¹, MIRCEA RUBA¹, (Member, IEEE),
RAUL OCTAVIAN NEMES¹, (Member, IEEE), AND
CLAUDIA MARTIS¹, (Senior Member, IEEE)

Department of Electrical Machines and Drives, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania

Corresponding author: Maria Raluca Raia (raluca.raia@emd.utcluj.ro)

This work was supported by the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, within PNCDI III, under Project PN-III-P2-2.1-PED-2019-4384 and Project PN-III-P1-1.1-TE-2019-0411.

ABSTRACT The present paper targets a solution for permanent magnet synchronous machine (PMSM) model order reduction (MOR) using artificial neural networks and machine learning techniques for data dimensionality reduction. The neural networks are trained using data obtained from a series of electromagnetic Finite Element Analysis (FEA), conducted in conditions imposed by the data dimensionality reduction method. The workflow proposed to build the PMSM MOR, starts with data generation, goes further to its post-processing, and finishes with the model training and experimental validation. In the study, data dimensionality reduction procedure (adaptive data generation) is performed to increase the computational efficiency, also maintaining the model accuracy. Different data reduction approaches are compared from the computational cost's point of view and their ease of use. The obtained results are compared to those obtained from FEA seeking the best solution for building the dynamic model. The resulting ROM is included in a real-time control prototyping platform to characterize machine's performances. The model accuracy and its usability are proved in a comparative analysis with simulated versus experimental measurements.

INDEX TERMS Artificial neural networks, adaptive data generation, machine learning, permanent magnet synchronous machine, dynamic simulation, real-time.

I. INTRODUCTION

In the automotive domain, the permanent magnet synchronous motor (PMSM) is preferred for both auxiliary and traction applications due to its well-known performances such as high power density and increased efficiency. The modelling process of the electrical machine using electromagnetic simulations based on Finite Element Analysis (FEA) represents the most accurate approach. However, its accuracy comes with the cost of increased computational time, even when parallelization techniques are involved, making it an inappropriate method for real-time applications. For the latter, besides accuracy, the computational time represents an important parameter. Hence, high fidelity motor

models need to be implemented in software packages that allow dynamic analysis. The reduced order model (ROM) is built using the data obtained from electromagnetic simulations, where only the relationship between the input and the output is taken into account and included in the dynamic model, usually in the form of multi-dimensional lookup tables (LUTs). The FEA motor model is seen like a black-box and it is employed in a series of simulations, resulting new computed data. The models with stored parameters into LUTs are dependent on the amount of the recorded data and on the machine phase current's range used to perform the electromagnetic simulations. In order to avoid the error introduced by the LUTs when extrapolating (i.e., during motor overload), the current's range is chosen to be a several times larger than the rated one, leading to increased time for data computation. Even more, the model's accuracy is highly influenced by the

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi¹.

current's range decimation; the smaller the current steps the higher the computational precision.

High-fidelity models can be achieved using modern methods, such as machine learning. The artificial neural network (ANN), that has an increased capacity to fit non-linear functions and to find patterns in the training data generated by the FEA model, can provide fast description of the system behaviour [1]. The implementation of an ANN consists in defining the model's inputs and outputs together with their connection given by the network's architecture and training algorithm. In the field of electrical machines, ANNs are implemented for machine control, anomaly detection and fault diagnosis, or for electromagnetic torque estimation, as presented in references [2]–[4]. A switched reluctance motor's (SRM) accurate model based on hybrid trained wavelet neural network (WNN) that combines genetic algorithm (GA) with gradient descent (GD) methods to train the network is proposed in [2]. In [3] and [4], torque estimation methods and machine control techniques based on neural network are detailed, reaching highly accurate results.

Furthermore, to reduce both computational and neural network training times, the data dimension used to train the network is minimized by applying machine learning dimensionality reduction techniques. Different methods for data reduction based on machine learning allow obtaining quickly high-accuracy models. Such an example is presented in [5], where a comparison of various data sampling and their effect on the accuracy of an ANN for torque estimation is discussed.

In this context, the paper proposes a flux-linkage quadrature (dq) based model order reduction (MOR) obtained by fitting two multi-layer artificial neural networks (ANNs). The novelty of this study rises from:

- 1) Involving data dimensionality reduction methods coming from the machine learning domain in the process of surrogate modelling. The reason behind this approach is that the PMSM's reduced order models developed using neuronal networks trained with reduced data sets are capable to reach high-accuracy results.
- 2) Describing the process of data computation and network training, performed in a significantly reduced amount of time, compared to the traditional approach.
- 3) Evaluating the efficiency of the dimensionality reduction methods by performing a comparison between the FEA, the experimental and the ANNs results obtained after training the network with different reduced data sets.
- 4) Quantifying the positive effect of data dimensionality reduction methods on time dedicated to extract the necessary data from electromagnetic simulations and on the time dedicated to train the networks.

The PMSM surrogate models discussed in this paper present reduced development time, although capable to offer accurate results including the possibility to predict the torque ripple.

The present paper is structured in the following sections: the machine used to validate the dynamic model and the machine governing equations are presented in Section II. In Section III, the architectures of the neuronal networks and the features of the training algorithm are discussed. The data dimensionality reduction procedures and their comparison in terms of accuracy and elapsed computational time are described in Section IV. The experimental validation of the theoretical studies is performed in Section V. Section VI is dedicated to final conclusions.

II. FLUX-LINKAGE-BASED PMSM MODEL

A. MACHINE UNDER STUDY

The development and validation of the dynamic PMSM model are based on the numerical analysis conducted on a eighteen slots and six poles PMSM, having the stator teeth surface shaped for torque and noise, vibration and harshness (NVH) characteristics improvement. The machine cross-section, its magnetic flux density and flux lines distributions are shown in Fig. 1. The main specifications of the machine under study are listed in Table 1.

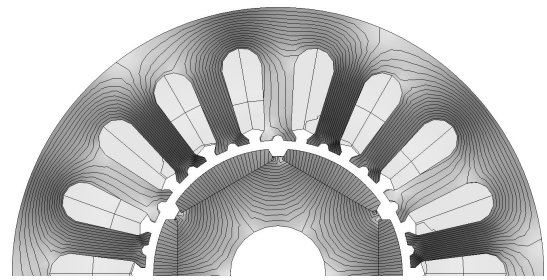


FIGURE 1. Cross-section, magnetic flux density and flux lines of the machine under study.

To construct the reduced order model of the machine under study, several 2D finite element analysis (FEA) based simulations are engaged at different rotor positions and armature currents. The quadrature model currents (d- and q- axis) are ranged from -200 [A] to 200 [A], with a step of 20 [A], while the rotor position is varied from 0 to 60 mechanical degrees.

TABLE 1. The main parameters of the machine under study.

Parameter	Unit	Value
Rated speed	rpm	1500
Rated voltage	V	50
Rated current	A	28
Rated torque	Nm	2
Output power	W	300
Stack length	mm	60

B. MACHINE MODELING

The general description of a PMSM is based on a system of equations referenced in a quadrature axes system, fixed to the rotor. This system is denoted dq0 and its corresponding

equations are listed in (1).

$$\begin{aligned} v_d &= R_s i_d + \frac{d\Psi_d}{dt} - \omega_r \Psi_q \\ v_q &= R_s i_q + \frac{d\Psi_q}{dt} + \omega_r \Psi_d \\ v_0 &= R_s i_0 + \frac{d\Psi_0}{dt} \end{aligned} \quad (1)$$

where v_d , v_q and v_0 are the machine voltages, i_d , i_q and i_0 are the machine armature currents, R_s represents the phases resistance, ω_r represents the angular speed and ψ_d , ψ_q and ψ_0 are the flux-linkage.

Considering that the supply system is perfectly balanced and the motor has star connection, the 0 components will be neglected from the dq0 equation presented in (1). Extracting from (1) the magnetic fluxes, their integral form results:

$$\begin{aligned} \psi_d &= \int (v_d - R_s i_d(\psi_d, \psi_q, \theta_r) + \omega_r \Psi_q) dt \\ \psi_q &= \int (v_q - R_s i_q(\psi_d, \psi_q, \theta_r) - \omega_r \Psi_d) dt \end{aligned} \quad (2)$$

where θ_r represents the rotor position.

The electromagnetic torque is usually computed as average value, hence no cogging or distortion due to the sinusoidal waveform of the magnetomotive force (MMF) due to the stator winding distribution are taken into account:

$$T_{em} = \frac{3}{2} p (\psi_d i_q - \psi_q i_d) \quad (3)$$

with p denoting the number of pole pairs.

However, to design a high-accuracy dynamic machine model, returning close to FEA and experimental results, the cogging effect and the MMF distortion must be considered. In doing so, the computation of the instantaneous torque will include its ripples. This has to be solved without additional calculations. Hence, extracting from FEA the torque function of the d- and q- axis currents and the rotor position should be enough. From the same FEA model the flux-linkage values are extracted, used also for building the dynamic PMSM model.

III. ARTIFICIAL NEURAL NETWORK

The artificial neural networks (ANN) is a machine learning approach that mimics the human intelligence and has a non-linear fitting capacity, which makes it a good candidate for modelling reduced-order electrical machines [2]. There are two basic ANN architectures: the single-layer ANN, where the input neurons are directly linked to the output neurons, and the multi-layers ANN, where the input neurons are separated from the output ones by a hidden layer[1]. Even if a network can have multiples hidden layers, no more than three are commonly used [6].

In order to obtain accurate results, the ANN architecture chosen to be implemented is a multi-layer network, consisting in an input layer that transmits and distributes the information to all the neurons of the intermediate layer, known as the hidden layer. Here, the data computation is performed and

passed to an output layer that constitutes the model result. Therefore, the considered network is a feed-forward one, as the data is transmitted only from the left layer to the right layer without feed-back connections. Furthermore, the nodes of the same layer are not interconnected, but these are linked to the nodes of the following layer, as depicted in Fig. 2. This hidden layer handles data manipulation coming from the input nodes by modifying specific parameters (i.e. weights and biases) that describe connections of adjacent layers.

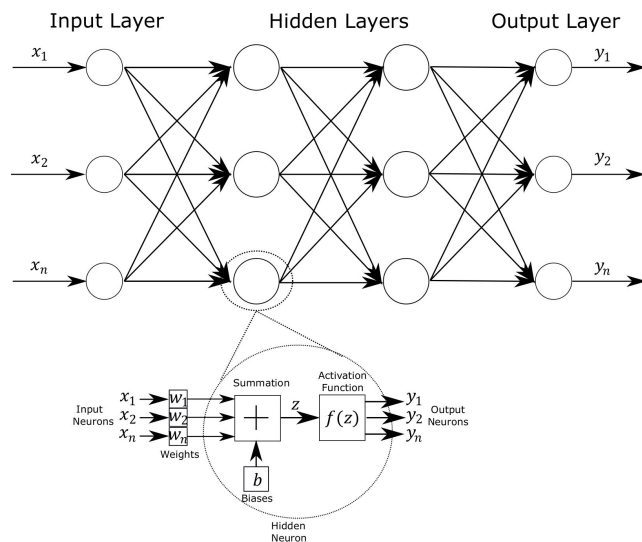


FIGURE 2. Neural network structure [7].

Fig. 2 depicts the process where the information coming from the input neurons, x_i , is multiplied with weights elements and then summed with bias ones. The result of this process, performed in the hidden neurons, is expressed as:

$$z_j = \sum_j^n \sum_i^n w_{ij} x_i + b_{ij} \quad (4)$$

where the terms w_{ij} and b_{ij} represent the weight and the bias between the i^{th} input neuron and the j^{th} hidden neuron. n denotes the number of inputs.

The resulting input, z , is subject to a transfer function computing the neural network's output. A non-linear activation function, the sigmoid function, is adopted in (5).

$$f(z) = \frac{1}{1 + e^{-z}} \quad (5)$$

A. MODEL IMPLEMENTATION

In order to create the reduced order model of the machine under study, two ANNs are trained. The first one deals with current estimation from the flux-linkage and rotor position, while the second one estimates the torque based on the d- and q-axis currents and rotor position. The block diagram of the motor model developed in dq reference frame is presented in Fig. 3.

The flux-linkage network has a structure composed of four layers with 3 input neurons, 60 ones in the first hidden layer,

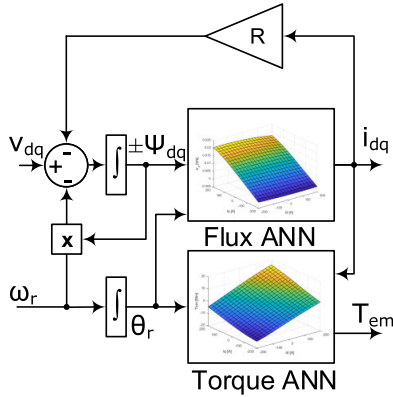


FIGURE 3. Block diagrams for the motor dq reference frame model.

20 in the second hidden layer and 2 in the output layer. The network is trained by imposing as inputs the d- and q-axis flux-linkages and the rotor position, resulted from FEA, outputting the d- and q-axis currents. The connection between the input and output layers is assured by the hidden layers. Comparing the look-up-tables (LUTs)-based model and the ANN-based motor model, the later needs less data processing time. If the network is optimally trained and returns accurate results, the process of flux-linkage inversion performed for LUTs-based models, is avoided. The mathematical inversion of 3D LUTs is known to be time consuming, reaching the elapsed time of 1-2 hours. Hence, if the inversion is eliminated, the overall process is solved faster, reaching a good correlation between the input and the output within the training process.

The network trained for torque estimation has a simple structure, with only three layers. The input layer consists of 3 neurons, the hidden layer has 100, while the output one is composed by a single neuron. The training process is performed by imposing the i_d , i_q and the rotor position, θ_r , as inputs and torque values as targets.

B. LEVENBERG-MARQUARDT BACKPROPAGATION ALGORITHM

To train the developed multi-layer feed-forward ANN, the Levenberg-Marquardt backpropagation algorithm (LMBPA), is used. This is derived from the Newton algorithm for non-linear functions minimisation [8]. Considering that the Newton method for minimising functions uses the function Hessian matrix ($\nabla^2 F$) and function gradient matrix (∇F), the weights and biases are updated by the following the rule:

$$w_{k+1} = w_k - (\nabla^2 F)^{-1} \nabla F_k \tag{6}$$

Assuming that the function F , desired to be minimised with respect to a vector parameter (in this case, weight vector), can be expressed as the sum of square errors, the Levenberg-Marquardt algorithm is minimising the error

function, E [9]:

$$E(x, w) = \frac{1}{2} \sum_{t=1}^T \sum_{n=1}^N e_{t,n}^2 \tag{7}$$

Here x is the input, w is the weight vector, t the training sample and $e_{t,n}^2$ is the error resulted for the training sample t at output n . The difference is quantified as the absolute error between the output reference (expected value) and the obtained one, as follows:

$$e_{t,n} = y_{t,n}^* - y_{t,n} \tag{8}$$

The Newton algorithm computes the minimum of the error function using the Hessian matrix, obtained from the function's second derivative:

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ \dots & \dots & \dots \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & \dots & \frac{\partial^2 E}{\partial w_n^2} \end{bmatrix} \tag{9}$$

The Levenberg-Marquardt algorithm approximates the Hessian matrix from the Jacobian one, J , a parameter μ and the identity matrix, I , as it is described in (10). It is worth mentioning that if μ takes small values, the algorithm transforms into Gauss-Newton algorithm.

$$H = \nabla^2 E = J^T J + \mu I \tag{10}$$

where the Jacobian matrix is expressed as:

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \dots & \frac{\partial e_1}{\partial w_n} \\ \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \dots & \frac{\partial e_n}{\partial w_n} \end{bmatrix} \tag{11}$$

Calculating the gradient function using its Jacobian, reduces the computation time:

$$\begin{aligned} g_i &= \frac{\partial E}{\partial w_i} = \frac{\partial (\sum_{t=1}^T \sum_{n=1}^N e_{t,n}^2)}{\partial w_i} \\ &= \sum_{t=1}^T \sum_{n=1}^N \frac{\partial e_{t,n}}{\partial w_i} e_{t,n} = J e \end{aligned} \tag{12}$$

Substituting (10) and (12) in (6), the Levenberg-Marquardt algorithm weights updating rule is [9]:

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \tag{13}$$

C. NEURAL NETWORK OVERFITTING

One of the main challenges when training a neural network is to quantify the required invested effort to reach an optimal level. Two important situations appear when training the neuronal networks: underfitting, when the network is not trained enough and is not reaching satisfactory results, and the overfitting. The latter issue appears when the network is trained to perfectly fit the training data by learning its particularities. In doing so, its capability to generalize and

adapt to new input data is decreased. Hence, even if the network offers very good results for particular data, it will prove poor performance in case of new inputs.

A solution to this problem may be to train multiple networks with different parameters (weights, biases) and different data partitioning into training, validation and test samples, until an adequate generalization is reached. In doing so the model is capable to predict other data. Another approach to avoid the overfitting drawbacks is the “early stopping” method. Using this method, the data is divided, based on their destination, in training data, used to compute the network weights and biases, validation and testing data. In parallel, the errors coming from the validation and training are evaluated in order to analyse the generalisation capacity of the network. Normally, the errors from training and validation decrease during the training process. When the network starts to overfit, the validation error increases and the training is stopped [10].

However, the validation error can increase during a number of successive iterations, hence a stopping criterion must be included in the algorithm [10].

To avoid the overfitting issue, for the implemented neural networks, the available data is divided in 70 [%] for the training process, 15 [%] for the validation and 15 [%] for testing. Moreover, the training process is stopped when the validation error keeps increasing during 300 successive iterations. The weights and biases, computed for the minimum validation error, are returned. The generalization capability of the developed neural networks is tested by imposing new, unseen inputs and comparing the output with results obtained from FEA simulations.

D. FLUX-LINKAGE VS. LINEAR ANN-BASED MODELS

To prove the necessity of building a flux-linkage-based model, where both flux-currents and currents-torque relationships are modeled with ANN structures, the model proposed in Fig. 3 is compared with a linear model. The linear model is an inductance-based one that relies on the linear relationship between the fluxes and the self and mutual inductances. Assuming that the three-phase supply currents system is perfectly balanced and the motor has a star winding connection, the linear model can be expressed by the voltage equations, where the state variables are represented by the quadrature currents:

$$\begin{aligned}
 v_d &= R_s i_d + L_d \frac{di_d}{dt} + L_{dq} \frac{di_q}{dt} - \omega_r L_q i_q \\
 v_q &= R_s i_q + L_q \frac{di_q}{dt} + L_{qd} \frac{di_d}{dt} + \omega_r L_d i_d + \psi_{md} \omega_r
 \end{aligned}
 \tag{14}$$

where v_d and v_q are the machine voltages, L_d and L_q represent the d- and q-axis inductances, L_{dq} and L_{qd} are the cross coupling inductances, ω_r is the angular speed and ψ_{md} is the permanent magnet flux.

The motor parameters (i.e., quadrature inductances) are obtained as a function of d- and q-axis currents from

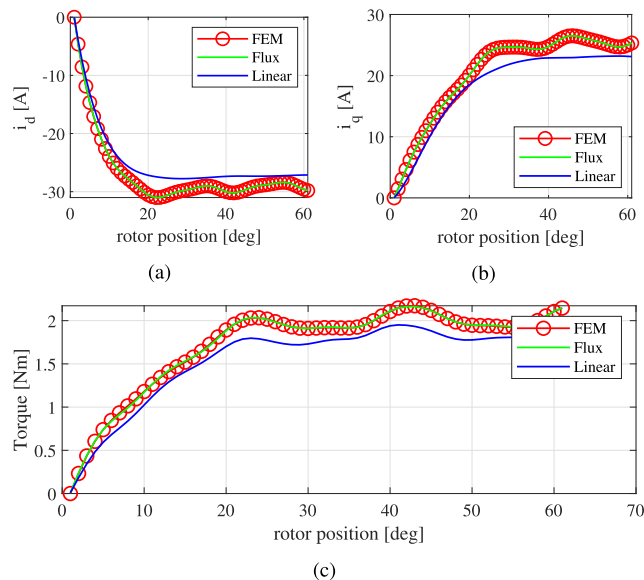


FIGURE 4. Flux-linkage and linear ANN-based models (a) d-axis current (b) q-axis current and (c) electromagnetic torque waveforms.

FEA using frozen permeability method (FPM) [11], [12]. The inductances are included in the dynamic model using 2D LUTs.

Furthermore, in order to capture the torque ripple, the linear model uses for electromagnetic torque prediction the Torque ANN with a three layers structure described in Section III-A.

The accuracy of the flux-linkage model presented in this paper is tested by comparing its results with those from the linear and from FEA-based models. Hence, all three models are fed by a sinusoidal 3-phase voltage system:

$$v_k = v_d \cos(\theta_r + \alpha) - v_q \sin(\theta_r + \alpha)
 \tag{15}$$

where $k \in \{a, b, c\}$ and $\alpha \in \{0, 2\pi/3, -2\pi/3\}$.

Both the torque and d- and q-axis currents obtained for $v_d = 0[V]$ and $v_q = 4.4[V]$, when the rotor is turning at a constant speed equal to 1500 [rpm] are shown in Fig. 4. The presented results show the simulations of the flux-linkage and linear ANN-based models compared to FEA ones. Even if the linear model doesn't require time for training a neural network or for LUTs inversion, the obtained results are less precise. The electromagnetic torque waveforms are presented in Fig. 4 c). It can be observed that despite the fact that both flux-linkage and linear models use a neural network for torque prediction, the linear model is not able to replicate the FEA values. This is because the linear model is not able to accurately compute the armature currents, as the voltage equations are linear modeled. The flux-linkage ANN-based model is able to accurately predict the armature currents using the Flux ANN structure and torque values using the Torque ANN structure. Therefore, the flux-linkage ANN-based model will be used for the next studies presented in this paper.

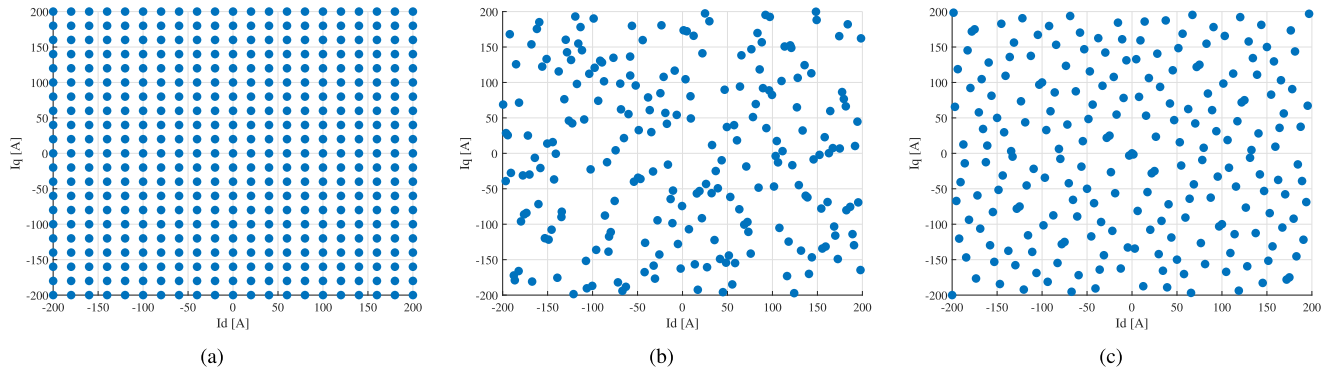


FIGURE 5. (a) Equally distributed data, EDD, (441 points), (b) LHS data distribution (220 points) and (c) Sobol sequence (220 points).

IV. DATA DIMENSIONALITY REDUCTION

Traditionally, the data used for electrical machines model order reduction is non-adaptive. This means the data is extracted from electromagnetic simulations with imposed conditions, usually defined by a fixed number of d- and q-axis currents and rotor positions. These are splitting the input space in equal intervals, as depicted in Fig. 5 a), where a number of 441 i_d and i_q combinations, for every rotor position, are used to train the neural network. The data is extracted and stored in LUTs, or, in our case, the data is used to train neural networks meaning that the information is computed in a single step and the amount of data generated may be more than the network requires in order to reach high fidelity results.

To overcome this issue, a machine learning method was implemented to reduce the data generated dimension used to train the proposed neural networks.

A. ADAPTIVE DATA GENERATION PROCESS

The active learning method used to reduce the data dimension is the adaptive samples generation. Using this technique, the data necessary for network training is computed and extracted in multiple steps. Firstly, a small amount of electromagnetic simulations are conducted in order to compute and extract the necessary data to train the network and evaluate the output error. If this is under a desired threshold, it is concluded that the network is sufficiently trained and the process is stopped. If the network doesn't behave as it is expected, additional data computation is performed. The process consisting in data generation, neural network training and error evaluation is recurrent and it stops when the network reaches a desired accuracy.

For the adaptive data generation process, the input samples are created using a pseudo-random algorithm that fills the input space uniformly, avoiding clusters and gaps within data. In [13] three sampling methods, Monte Carlo (MC), Latin Hypercube Sampling (LHS) and Sobol sequence, are described and their performances are compared. The Sobol sequence proves the best points distribution in n-dimensions. As it can be observed in Fig. 5, sampling based on Sobol sequence is characterised by increased uniformity of the input

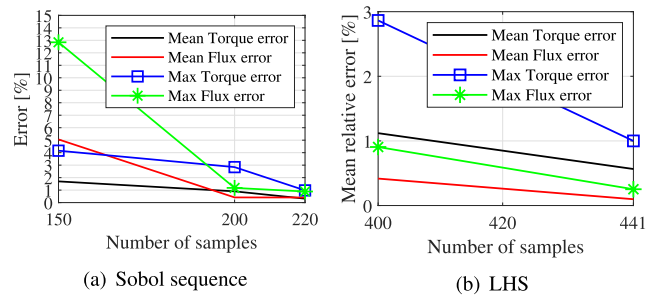


FIGURE 6. Networks training relative errors expressed in [%].

points, compared with the LHS method. The main features of the Sobol sequence are:

- 1) An increased space distribution as the number of points, N , goes to infinite.
- 2) A satisfactory distribution for N goes to zero.
- 3) A fast converging algorithm.

B. IMPLEMENTATION

To maintain the ANNs results relative error under 1 [%], the FEA simulations are performed for a number of input samples obtained using both LHS and Sobol sequence methods. Hence, starting from 100 i_d - i_q samples for every rotor position, their number is increased until the targeted accuracy is reached. The errors obtained using the Sobol sequence method decrease once the number of i_d - i_q combinations is increased, a satisfactory error being obtained for more than 200 i_d - i_q input samples. As depicted in Fig. 6 a), the mean relative error obtained from both torque (represented with black line) and flux networks (represented with red line) is small for 200 i_d - i_q training samples. Even if the Flux ANN maximum error (marked with green line) is less than 1 [%], the Torque ANN maximum relative error is 2.84 [%], exceeding the targeted value. The maximum and mean relative errors of the Torque ANN decrease under 1 [%] when the input is increased with 20 samples. In doing so, the flux ANN errors are slightly decreased with 220 i_d - i_q input combinations.

Therefore, by generating data through Sobol sequence method, the number of input samples for torque and

flux-linkage neural network training is reduced to 220 points. Comparing this with the classical data generation method, Equally Distributed Data (EDD), presented in Fig. 5 a), where the input consists of 441 samples, proves that the data dimension is reduced with 50 [%]. The amount of data computed for neural networks training is reduced significantly. This also has a positive impact on the elapsed time for FEA and training, as shown in Table 2. Both time periods, dedicated to FEA and ANN training are decreased, with up to 50 [%], compared to the case when data is extracted for an equally distributed input samples. Moreover, a comparison between the training time obtained for different number of samples can be identified in Fig. 7. New training data is generated using FEA simulations, starting from a set of 150 samples up to 441 samples, with variable step. It can be observed that the training time increases significantly once the number of samples is expanded. The training process for a Flux ANN is highly time consuming due to its elaborated structure.

TABLE 2. The time dedicated to generate the data and train the ANNs.

Data type	Nb. samples	FE Analysis	ANN training	Total time
EDD	441x61	558 [min]	90 [min]	648 [min]
LHS	400x61	506 [min]	70 [min]	576 [min]
LHS	441x61	558 [min]	75 [min]	633 [min]
Sobol	150x61	190 [min]	20 [min]	210 [min]
Sobol	200x61	253 [min]	39 [min]	292 [min]
Sobol	220x61	279 [min]	46 [min]	325 [min]

The results obtained for an artificial network trained with data from FEA at imposed input points generated with Sobol sequence method are accurate enough with reduced training time. On the other hand, the neural networks described in Section III-A, trained with data from FEA with inputs generated with LHS method start to be accurate enough above 400 input samples. This is a direct consequence of its uneven filling pattern of the input space. No extreme sample points are taken into account, as observed in Fig. 5 c).

Figure 6 b) presents the relative errors obtained from the Torque and Flux ANNs trained with LHS data. The Torque ANN mean relative error for 400 samples is 1.1 [%] and the maximum error is 2.9 [%], exceeding the reference (i.e., 1 [%]). Hence, the number of samples is increased to 441, decreasing the errors under the desired limit. Even if the networks trained with data generated using LHS method are accurate enough for 441 used samples, the dimension of the input space is larger, compared with the EDD data and Sobol sequence method.

A good agreement between FEA results, the networks sequentially trained with full data dimension (EDD), reduced data dimension using Sobol sequence (220 samples) and LHS (441 samples) are identified in Fig. 8 a) and b). The torque waveforms in steady state at rated speed are compared in Fig. 8 a). In Fig. 8 b), the instantaneous errors are depicted assuming the FEA results as references. It can be observed that the Sobol 220 network computes the results closest to FEA. This is a consequence of its capacity to uniformly fit the input space samples.

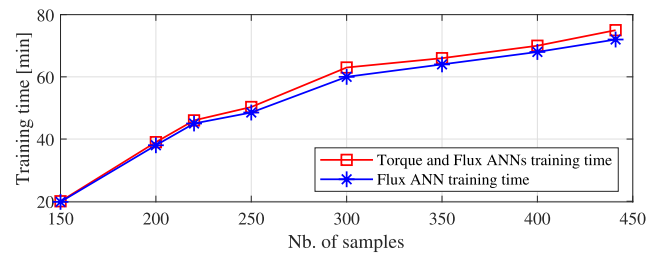


FIGURE 7. The time dedicated to train the ANNs.

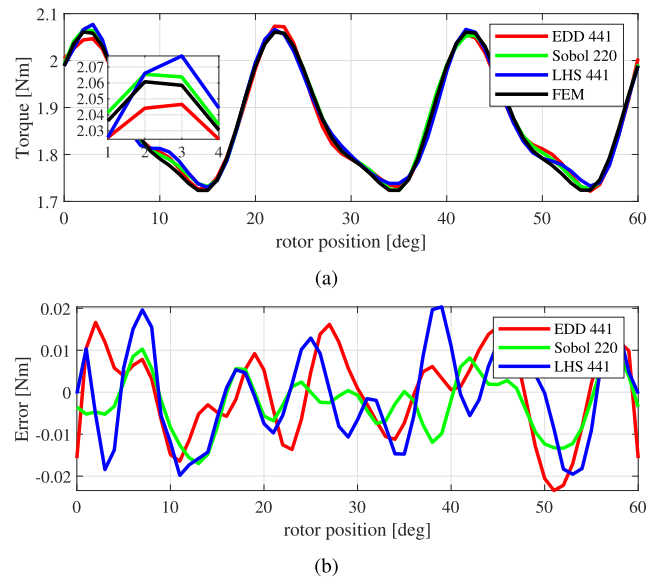


FIGURE 8. (a) electromagnetic torque and (b) instantaneous error of the FEM and ANN models obtained for different data samples distribution, at steady state.

C. SOBOL 220 NETWORK MODEL VALIDATION

To prove the accuracy and capacity to predict nonlinear highly saturated machine behaviour of the ANNs trained with Sobol 220 sequence, its results are compared to those from FEA. Firstly, the validation is performed at rated current and constant rated speed. One can observe in Fig. 9 a) that the Sobol 220 network model is able to replicate the torque ripple obtained from FEA simulation. Its harmonic content is analysed in Fig. 9 b). It can be observed that for rated current, the neural network model's result is similar to the one from FEA.

In the second and third study case, the machine is saturated by increasing the phase current by five and eight times the rated value. It is worth mentioning that the latter condition is imposed in order to prove the neural network model's prediction capacity. The current value (i.e., eight times rated current) is new, previously unseen, meaning that it was not included in the current range for which the data was extracted from FEA simulations. The torque obtained for five and then for eight times the rated current are depicted in Fig. 10 a) and Fig. 11 a). In both cases, a good correlation with FEA results is identified, with an error under 1 [%]. Furthermore, it can

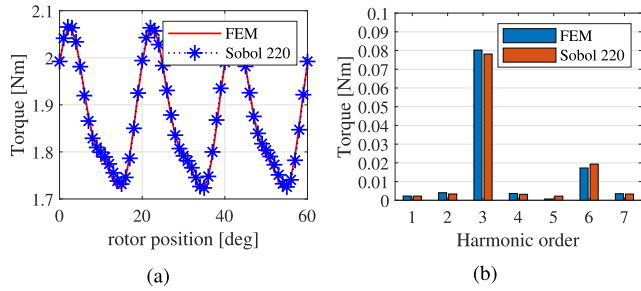


FIGURE 9. (a) FEM vs ANN Torque comparison obtained for rated current and (b) Frequency spectrum of the electromagnetic torque.

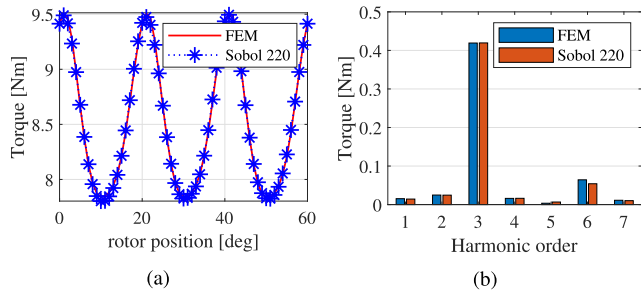


FIGURE 10. (a) FEM vs ANN Torque comparison obtained for 5 times rated current and (b) Frequency spectrum of the electromagnetic torque.

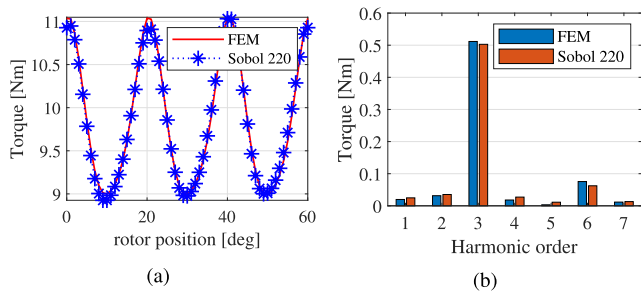


FIGURE 11. (a) FEM vs ANN Torque comparison obtained for 8 times rated current and (b) Frequency spectrum of the electromagnetic torque.

be observed that even when the machine is highly saturated, the dynamic model is able to replicate the torque ripples created by the distorted magnetic field and cogging effect. The harmonic content of the resulted torques are presented in Fig. 10 b) and Fig. 11 b). Although the amplitude of the harmonics is changed, compared to the case of rated current, the harmonic content of the neural network model is similar to the one obtained with FEA. Therefore, the Sobol 220 ANN model can accurately predict the electromagnetic torque under rated or increased saturation conditions and will be compared with the experimental measurements.

V. EXPERIMENTAL VALIDATION

To prove the performances and the accuracy of the model obtained using the methods presented in the previous sections, the results of the dynamic model are compared with experimental ones. As detailed previously, the dynamical simulation model was built in Matlab/Simulink environment. In Section IV, a comparative analysis was presented, super-

imposing results from FEA and Simulink models for the same machine, operated in the same conditions. At that point, the high accuracy of the PMSM’s Simulink model using the Sobol 220 networks for current and torque estimation was proved. This was considered a preliminary validation step, handshaking the FEA with the Simulink model.

The dynamic model was built using the equations detailed in Section II, while the flux-linkages for the dq axes and torque values were recorded and used to train the neural networks. The PMSM’s simulation model is presented in Fig. 12, as general organization method of such flux-linkages-based analysis program. The initial conditions of the two integrators included in the dynamic model are set to be equal with the quadrature values of the flux-linkage obtained at no-load condition, for the initial rotor position.

As a PMSM requires electronic control to follow imposed references, the authors attached to the PMSM model, in Matlab/Simulink, the Field Oriented Control (FOC). This computes the necessary voltages to be supplied to the machine to ensure it pursuits the imposed speed reference independent of its load torque. The interest of this study is not focused on detailing the FOC, the reader being encouraged to visit references [14] and [15].

What remains to be proved is the validity of the dynamic machine model. This should return the same results like the actual machine experimentally tested. By this, the results of the simulation are proved to overlap those obtained on the actual test bench. If this target is reached, one can consider that all models, FEA, dynamic and real one behave identically, insuring high accuracy simulation results.

In order to perform a comprehensive analysis, the model is referenced with random speed and load torque variations.

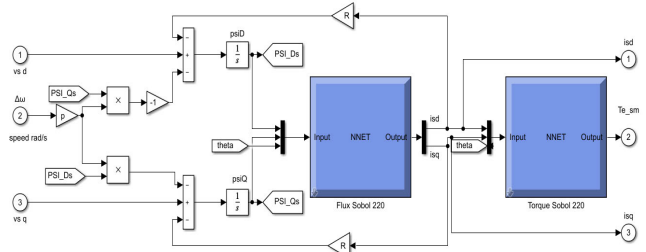


FIGURE 12. The Matlab Simulink PMSM model.

These are used for the laboratory testing of the actual PMSM versus the modelled one. The setup of the laboratory test-bench is depicted in Fig. 13, including: the PMSM under study and its 3-phase inverter, a DC machine as load machine connected to a programmable load, a currents measurement unit and a NI RSeries FPGA used as general controller.

To avoid redundancy, the results will be presented in a comparative analysis superimposing the references with the data from the simulation model and the one from the test-bench.

In Fig. 14, speed and torque variations are compared to the references. One can observe that there is a quite good agreement between the plots. It has to be mentioned that the

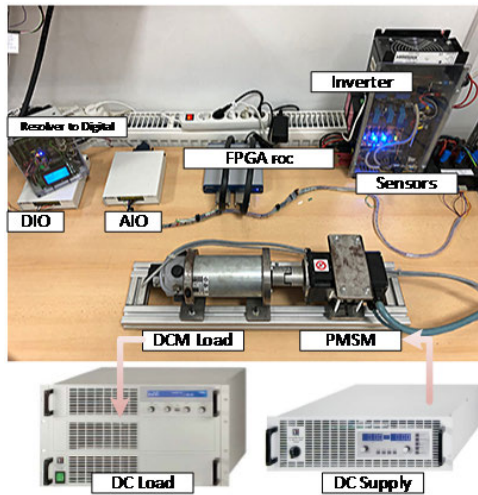


FIGURE 13. The laboratory test-bench setup.

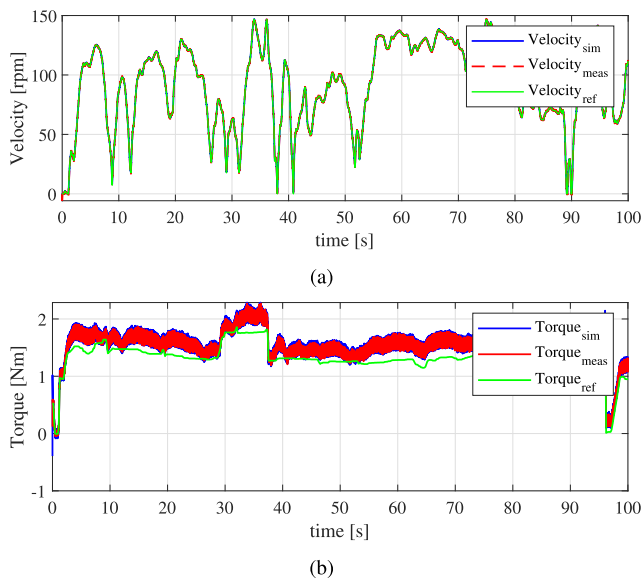


FIGURE 14. Simulated, measured and reference speed (a) and torque (b) variations.

measured values were not filtered by any means in order to have a realistic comparison of the raw data versus the one obtained via simulations. It must be explained that the generated torque is larger than the reference one because the machine has to overcome, besides the load torque, the friction forces that appear during its operation. It can be observed in Fig. 13 that the test-bench does not include a torque meter. However, the authors identify the PMSM torque using the dq currents and the machine parameters, based on (3). The PMSM used is a custom-made machine and for this reason the FEA model is available in detail for the authors. The torque from the FEA model, the test-data from the machine's manufacturer and the torque estimated by authors agree. Hence, it is considered a 3-way validation assuring the authors that the analytical estimation returns the actual electromagnetic torque.

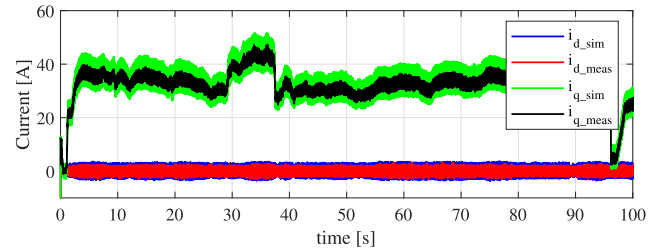


FIGURE 15. Simulated vs. measured dq machine currents.

Going further with the comparison, in Fig. 15 the i_d and i_q currents obtained from the simulation are compared to those measured on the test-bench. Again, the good agreement proves that the simulation model performs precisely like the test-bench.

The random nature of the speed and load torque reference profiles ensures coverage of a large range of possible machine conditions. The results presented in this section prove the accuracy of the simulation program compared to their measured homologous. However, not the simulation program itself is the center of interest but the concept of engaging machine learning techniques to create highly precise data necessary to describe the PMSM. Actually, by these results it is proven that using such a philosophy when generating the model data, one can build a simulation program that performs just like its real twin, using reduced data quantity.

VI. CONCLUSION

This paper presented a PMSM modelling approach characterised by combining artificial neural networks and machine learning techniques for data dimensionality reduction. Two artificial neural networks were used for current and torque prediction, proving increased capacity to fit the data extracted from FEA that were performed accordingly with the data reduction methods. To reduce the computational and networks training time, two adaptive data generation techniques, the Latin Hypercube Sampling and the Sobol sequence were implemented and described. The results (i.e., electromagnetic torque and armature currents) obtained from ANNs trained with different data sets were compared with the ones from FEA and the accuracy and the capacity to decrease the computational effort of the proposed networks was confirmed.

Comparing the results obtained from the dynamic models with the FEA ones, it was concluded that the network trained with 220×61 input combinations generated with Sobol sequence has the best performances, reducing both computational time and data dimension with 50[%], keeping the results relative error under 1[%]. Therefore, the 220 Sobol network was included in the final motor model dedicated to real-time platform integration.

An experimental validation of the PMSM model was carried out by comparing results from the simulations with those obtained in the laboratory. These proved the accuracy of the simulation model and by this scientific value is added to the study showing the benefits of engaging such methods when it comes to high performance simulation programs.

ACKNOWLEDGMENT

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2019-4384, within PNCDI III. This work was also supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, project number PN-III-P1-1.1-TE-2019-0411, within PNCDI III.

REFERENCES

- [1] C. C. Aggarwal, *Neural Networks and Deep Learning*. New York, NY, USA: Springer, 2018.
- [2] S. Song, L. Ge, S. Ma, and M. Zhang, "Accurate modeling of switched reluctance machine based on hybrid trained WNN," *AIP Adv.*, vol. 4, no. 4, Apr. 2014, Art. no. 047130.
- [3] Y.-B. Yan, J.-N. Liang, T.-F. Sun, J.-P. Geng, Gang-Xie, and D.-J. Pan, "Torque estimation and control of PMSM based on deep learning," in *Proc. 22nd Int. Conf. Electr. Mach. Syst. (ICEMS)*, Aug. 2019, pp. 1–6.
- [4] M. M. Saafan, A. Y. Haikal, S. F. Saraya, and F. F. Areeed, "Artificial neural network control of permanent magnet synchronous motor," *Int. J. Comput. Appl.*, vol. 37, no. 5, pp. 9–18, Jan. 2012.
- [5] M. Tahkola, J. Keranen, D. Sedov, M. F. Far, and J. Kortelainen, "Surrogate modeling of electrical machine torque using artificial neural networks," *IEEE Access*, vol. 8, pp. 220027–220045, 2020.
- [6] M. Kubat, *An Introduction to Machine Learning*, 2nd ed. Springer, 2017.
- [7] S. Haykin, *Neural Networks and Learning Machines*, vol. 3, 3rd ed. London, U.K.: Pearson, 2008.
- [8] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
- [9] H. Yu and B. M. Wilamowski, "Levenberg-Marquardt training," in *The Industrial Electronics Handbook*, B. M. Wilamowski and J. D. Irwin, Eds. Boca Raton, FL, USA: CRC Press, Feb. 2011. Accessed: Mar. 22, 2021.
- [10] L. Prechelt, *Early Stopping—But When*, G. B. Orr and K.-R. Müller, Eds. Berlin, Germany: Springer, 1998.
- [11] N. Bianchi and S. Bolognani, "Magnetic models of saturated interior permanent magnet motors based on finite element analysis," in *Proc. IEEE Ind. Appl. Conf.*, vol. 1, Oct. 1998, pp. 27–34, doi: [10.1109/IAS.1998.732255](https://doi.org/10.1109/IAS.1998.732255).
- [12] D. E. Pinto, A.-C. Pop, J. Kempkes, and J. Gyselinck, "Reduced-order eddy-current loss modelling of electrical machines using lookup tables obtained via 2D finite-element," in *Proc. 13th Int. Conf. Electr. Mach. (ICEM)*, Sep. 2018, pp. 386–392.
- [13] S. Kucherenko, D. Albrecht, and A. Saltelli, "Exploring multi-dimensional spaces: A comparison of latin hypercube and quasi Monte Carlo sampling techniques," 2015, *arXiv:1505.02350*. [Online]. Available: <http://arxiv.org/abs/1505.02350>
- [14] M. Ruba, R. O. Nemes, S. M. Ciomei, C. Martis, A. Bouscayrol, and H. Hedeşiu, "Digital twin real-time FPGA implementation for light electric vehicle propulsion system using EMR organization," in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Oct. 2019, pp. 1–6, doi: [10.1109/VPPC46532.2019.8952428](https://doi.org/10.1109/VPPC46532.2019.8952428).
- [15] S. Ciomei, R. Nemeş, M. Ruba, C. Martiş, and H. Hedeşiu, "Performance analysis of urban light electric vehicle propulsion system's multi-level modelling," in *Proc. IEEE Int. Conf. Autom., Qual. Test., Robot. (AQTR)*, May 2020, pp. 1–6, doi: [10.1109/AQTR49680.2020.9129938](https://doi.org/10.1109/AQTR49680.2020.9129938).



MIRCEA RUBA (Member, IEEE) received the Ph.D. degree in electrical engineering from the Technical University of Cluj-Napoca, Romania. The topic of his Ph.D. thesis was to develop a fault tolerant modular switched reluctance machine. His research interests include power electronics, electrical machines, fault tolerance, batteries, and electric vehicles. He is currently a Lecturer with the Technical University of Cluj-Napoca. He has published more than 90 conference papers, five books, six patents, and membered more than 20 research grants, such as FP7 and H2020. He is a book editor with Intech and reviewed many articles of journals from the IEEE TRANSACTIONS.



RAUL OCTAVIAN NEMES (Member, IEEE) was born in Zalău, Sălaj, Romania, in 1993. He received the B.S. and M.S. degrees in electrical engineering from the Technical University of Cluj-Napoca, Cluj-Napoca, in 2018, where he is currently pursuing the Ph.D. degree.

Since 2001, he has been a Researcher with the Department of Electrical Machines and Drives, Technical University of Cluj-Napoca. He is the author of one book chapter, more than 15 articles,

and one journal article. His research interests include electric drives, energy storage systems, and power electronics.



CLAUDIA MARTIS (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in electrical engineering from the Technical University of Cluj-Napoca, Romania, in 1990 and 2001, respectively. Since 1996, she has been a member of the teaching staff with the Faculty of Electrical Engineering, Technical University of Cluj-Napoca. She is currently a Professor with the Department of Electrical Machines and Drives, Technical University of Cluj-Napoca. She has published more

than 120 scientific articles, (more than 60 in ISI and IDB journals and conference proceedings) and five books, three of them as a first author. She was a coordinator of 11 national and seven European projects. All the projects targeted automotive applications of electrical machines and drives. Her research interests include electrical machines and drives design, modeling, analysis, and testing for automotive, and renewable energy-based and industrial applications.



MARIA RALUCA RAIA received the B.Sc. and M.Sc. degrees in electrical engineering from the Technical University of Cluj-Napoca (TUCN), Romania, in 2018 and 2020, respectively, where she is currently pursuing the Ph.D. degree. She is currently a Research Assistant with the Department of Electrical Machines and Drives, Faculty of Electrical Engineering, TUCN. Her research interests include system-level simulations, multi-physical modeling, reduced order modeling of electrical machines, NVH, and electric vehicles.