Received: 28 December 2023 | Revised: 26 February 2024 | Accepted: 29 February 2024

DOI: 10.1002/jnm.3226

INVITED PAPER

Neural network-based simulation of fields and losses in electrical machines with ferromagnetic laminated cores

Florent Purnode | **François Henrotte** | **Gilles Louppe** | **Christophe Geuzaine**

Department of Electrical Engineering and Computer Science, University of Liège, Belgium

Correspondence
Corresponding author Christophe Geuzaine, Email: cgeuzaine@uliege.be

Abstract

Due to the distribution of eddy currents inside ferromagnetic laminations, the accurate modeling of magnetic fields and losses in the laminated cores of electrical machines requires resolving individual laminations with a fine 3D discretization. This yields finite element models so huge and costly that they are unusable in daily industrial R&D. In consequence, hysteresis and eddy currents in laminations are often simply disregarded in the modeling: the laminated core is assumed to be made of a reversible (non lossy) saturable material, and magnetic losses are evaluated a posteriori, by means of Steinmetz-Bertotti like empirical formulas. However, in a context where industry is struggling to minutely assess the impact of magnetic losses on their devices, this simplified approach is more and more regarded as inaccurate and unsatisfactory. This paper proposes a solution to this issue, based on homogenization and on detailed mesoscopic simulations of eddy currents and hysteresis inside the laminations. The proposed approach results in a close-to-conventional 2D magnetic vector potential finite element model, but equipped with an irreversible parametric material law to represent the ferromagnetic stack. In each finite element, the parameters of the law are obtained from a neural network trained to best fit the detailed mesoscopic simulations of the laminations subjected to the same local magnetic field. This way, all aspects of the irreversible ferromagnetic response are appropriately accounted for in the finite element simulation, but at a computational cost drastically reduced with regard to a brute force 3D calculation, and comparable to that of conventional 2D finite element simulations.

KEYWORDS
Magnetic hysteresis, Magnetic losses, Neural networks, Nonhomogeneous media

1 | INTRODUCTION

The detailed finite element (FE) analysis of magnetic losses in laminated-core electrical machines is a challenging task. This is due (i) to the inhomogeneous distribution of eddy currents and skin effect inside laminations, (ii) to the complexity of magnetic hysteresis, which demands a rather involved nonlinear irreversible model, and (iii) to the complex local interplay between these two phenomena. This holds particularly true in the context of state-of-the-art power electronics, due to increased operating frequencies and more distorted current and voltage waveforms [33].

The brute force modeling approach, that would consist in a 3D FE model with a fine discretization to resolve the field distribution inside individual laminations, is computationally extremely costly, and unsuitable for machine design. The laminated structure introduces however a symmetry in the problem that can be exploited to homogenize the stack thickness, reduce the 3D problem into a multiscale 2D problem and hence reduce the computational burden. Magnetic fields and losses across laminations are indeed ruled by a magneto-quasistatic model (with hysteresis and eddy currents) whose solution is in good approximation independent of the geometric details of the 2D electrical machine model. We call this lamination model the mesoscale[†] model, whereas the 2D machine model is dubbed the macroscale model.

In multiscale methods like the heterogeneous multiscale method (HMM) [31, 20] or the multilevel finite element method (FE2) [24], the information missing at the macroscale is generated during the simulation by solving the mesoscale model

[†] We use the term mesoscale, instead of microscale, because the characteristic thickness of a lamination is not in the microscopic range.

on a representative volume element, and by averaging its solution. But, as up-to-date homogenized mesoscale information is needed at each nonlinear iteration within each element and for each time step, and as this information not only consists in the constitutive relationship itself, but also in its gradients to construct a tangent stiffness matrix, the mesoscale model must be solved repeatedly a very large number of times. Multiscale approaches remain therefore too technically complex and still too computationally expensive for simulation tools to be used in routine electrical machine design.

In consequence, the practical solution in industrial R&D modeling nowadays is to disregard hysteresis and eddy currents in laminations during the FE simulation. Laminated cores are assumed to be made of a reversible (no hysteresis) and non-conducting (no eddy current) material, and magnetic losses are only evaluated *a posteriori*, by means of Steinmetz-Bertotti like empirical formulas [29, 5]. However, in a context where industry is struggling to minutely assess the impacts of magnetic losses on their devices, the *a posteriori* computation of losses is more and more regarded as oversimplified and unsatisfactory [27, 7, 3].

This paper proposes an efficient solution to this problem, building on the pragmatic two-step homogenization technique for ferromagnetic laminated cores previously proposed by some of the authors [12]. Instead of a direct multiscale coupling (like HMM or FE2 would do), the proposed approach makes use of an intermediary surrogate model to stand for the detailed mesoscale lamination model in the macroscale simulation. This surrogate model is a parametric analytic material law, whose parameter values are obtained in each finite element from a neural network (NN) trained to best fit the response of the mesoscale model given the same local magnetic field.

Over the recent years, machine learning techniques—and in particular NNs—have been used to address various aspects related to magnetic losses in electrical engineering and in power electronics. NNs of different sizes and complexities have been trained to provide a hysteresis model [1, 30], a soft magnetic composite core model [16], or even full machine models [25], with various assumptions and geometric simplifications. Mušeljić [18] uses a feed-forward NN to identify the parameters of the energy-based hysteresis model [11, 13].

In this paper, our objective is to offer a 2D homogenized FE model that accurately computes fields and losses in electrical machines with laminated stacks (motors, generators, transformers), but at a computational cost that remains compatible with the needs of parametric analysis in electrical machine design. This is made possible by the introduction of the parametric surrogate model, which takes the form of a conventional material law that has been generalized to accurately account for the irreversible behavior of the lamination stack, and whose parameters are locally obtained by the fast evaluation of a NN. This idea was first introduced in [22], with a simple training strategy and unidirectional fields. It is expanded in this paper up to the level of an efficient tool to be used in R&D applications.

The paper is organized as follows. The parametric surrogate model encapsulating the mesoscale effects in the lamination stack is introduced in Section 2. Section 3 then proceeds with the description of the NN architecture used to infer the surrogate model parameters based on time sequences of magnetic field values. The application of the overall workflow to the simulation of a reluctance machine is presented and analyzed in Section 4. General conclusions and perspective for future work are finally given in Section 5.

2 | PARAMETRIC IRREVERSIBLE MATERIAL LAW

Based on the pragmatic two-step homogenization technique for ferromagnetic laminated cores [12], we define the irreversible magnetic material law

$$\tilde{\mathbf{H}}\left(\mathbf{B},\dot{\mathbf{B}},p_k\right) = \underbrace{\left(p_0 + \left(p_1|\mathbf{B}|^2\right)^{p_2}\right)\mathbf{B}}_{\tilde{\mathbf{H}}_\mathrm{r}} + \underbrace{\overbrace{p_3\dot{\mathbf{B}}}^{\tilde{\mathbf{H}}_\mathrm{eddy}} + \overbrace{\frac{p_4}{\sqrt{p_5^2+|\dot{\mathbf{B}}|^2}}\dot{\mathbf{B}}}^{\tilde{\mathbf{H}}_\mathrm{hyst}}}_{\tilde{\mathbf{H}}_\mathrm{i}}, \tag{1}$$

referred from now on as the "p_k law". This law gives the magnetic field $\tilde{\mathbf{H}}$ as an analytic expression of the magnetic flux density $\mathbf{B}$, of its time derivative $\dot{\mathbf{B}}$ and of a set of six parameters p_k. The p_k law has been devised on basis of physical considerations. The magnetic field $\tilde{\mathbf{H}}$ is decomposed into a reversible part whose rate of magnetic work $\tilde{\mathbf{H}}_\mathrm{r}\cdot\dot{\mathbf{B}}$ integrated over a cycle is zero, and an irreversible part whose rate of magnetic work $\tilde{\mathbf{H}}_\mathrm{i}\cdot\dot{\mathbf{B}}$ integrated over a cycle is always positive. The reversible part of the magnetic field, $\tilde{\mathbf{H}}_\mathrm{r}$, is a polynomial anhysteretic saturation curve. Using a typical value of 5 for the exponent p_2, it exhibits a

saturation knee around $|\mathbf{B}|^2 = p_1^{-1}$. For small values of $|\mathbf{B}|$, $\tilde{\mathbf{H}}_\mathrm{r}$ is close to a straight line of slope $p_0 \approx \mu^{-1}$, with μ the magnetic permeability of the ferromagnetic sheets.

The second term, $\tilde{\mathbf{H}}_\mathrm{i}$, is associated with the irreversible phenomena: eddy currents and hysteresis. The eddy current term is inspired from the expression $\tilde{\mathbf{H}}_\mathrm{eddy} = \frac{\sigma d^2}{12}\dot{\mathbf{B}} \approx p_3\dot{\mathbf{B}}$, derived from the analytic calculation of eddy current losses in steel laminations [10], with σ the electrical conductivity of the ferromagnetic sheets, and d their thickness. A hysteresis loop, on the other hand, is characterized by a shift of the saturation curve to the right for ascending branches ($\mathrm{sign}(\dot{\mathbf{B}}) > 0$), and a shift to the left for descending branches ($\mathrm{sign}(\dot{\mathbf{B}}) < 0$). The hysteresis term, $\tilde{\mathbf{H}}_\mathrm{hyst}$, is thus represented by a regularized sign function ($\frac{x}{|x|} \approx \frac{x}{\sqrt{a^2+x^2}}$, with a a regularization parameter) of $\dot{\mathbf{B}}$ with a scaling factor p_4 equal to the coercive field H_c and a regularization parameter $p_5 \approx \omega B_\mathrm{sat}$, where ω is the angular frequency.

The main idea of the paper is to identify the values of the p_k parameters that best fit the response of a mesoscopic simulation of the lamination when it is subjected to the same magnetic field $\mathbf{H}$ as the one observed at macroscopic level. The purpose is thus to identify an abstract mapping $\mathbf{H} \mapsto p_k$. The details of the mesoscopic lamination model are described in Appendix A, and its response to a given excitation $\mathbf{H}$ is simply noted $\mathbf{B}(\mathbf{H})$ in what follows. With this notation, the mapping $\mathbf{H} \mapsto p_k$ to be identified can be expressed as the minimization problem

$$\mathbf{H} \mapsto p_k = \mathrm{argmin} \left\| \mathbf{H} - \tilde{\mathbf{H}}\big(\mathbf{B}(\mathbf{H}), \dot{\mathbf{B}}(\mathbf{H}), p_k\big) \right\|. \tag{2}$$

One focuses in this paper on the computation of magnetic losses in steady state. This means that the system shall be supplied with periodic voltages or currents. The time-stepping is then conducted over a sufficient number of periods to reach the steady state, and the solution over the last computed period is retained as the periodic solution of the problem. The periodic fields can thus be represented by arrays of $2N$ components containing their x and y components at the N timesteps of the period. We call these arrays *sequences*, and also note them $\mathbf{H}$, $\tilde{\mathbf{H}}$, $\mathbf{B}$ and $\dot{\mathbf{B}}$ with, e.g., $\mathbf{H} = \{H_{x,i}, H_{y,i}, i = 0, \ldots, N-1\}$, in order not to introduce another notation. The error to be minimized in (2) is chosen as

$$\left\|\mathbf{H} - \tilde{\mathbf{H}}\right\| = \sqrt{\frac{\sum_{i=0}^{N-1} \left(H_{x,i} - \tilde{H}_{x,i}\right)^2 + \left(H_{y,i} - \tilde{H}_{y,i}\right)^2}{\sum_{i=0}^{N-1} H_{x,i}^2 + H_{y,i}^2.}} \quad \text{with} \quad \tilde{\mathbf{H}} = \tilde{\mathbf{H}}\left(\mathbf{B}(\mathbf{H}), \dot{\mathbf{B}}(\mathbf{H}), p_k\right). \tag{3}$$

Three important remarks can be made about it. First, it is a relative error to put sequences with low and high amplitudes on a same footing. Second, the p_k law gives the magnetic field $\tilde{\mathbf{H}}$ as a function of $\mathbf{B}$, whereas the mesoscopic model gives $\mathbf{B}$ as a function of $\mathbf{H}$ to respect causality. The error (3) is thus not the comparison of the output of two concurrent models for the same input, as is usually the case. It is rather the comparison of the input $\mathbf{H}$ with the result of the chained application of the two models. Finally, one needs an input $\mathbf{H}$ sequence to perform the minimization (2). In practice, we obtain the input sequences by means of a preliminary simulation with the same mesh and with a conventional anhysteretic material law (see Section 4.1).

The performance of the p_k law, with properly identified parameters, is illustrated in Figure 1. The cycles (**a**) represent the pair $(\mathbf{H}, \mathbf{B}^\mathrm{prel})$, obtained from the preliminary simulation; the cycles (**b**) represent $(\mathbf{H}, \mathbf{B})$, where $\mathbf{B} = \mathbf{B}(\mathbf{H})$ is obtained by injecting the $\mathbf{H}$ sequence as an excitation into the lamination model; the cycles (**c**) represent $(\tilde{\mathbf{H}}, \mathbf{B})$, where $\tilde{\mathbf{H}}$ is the output of the p_k law with the p_k parameters obtained by solving (2) with $\mathbf{H}$ as input sequence, and with the global dual annealing optimization algorithm of `scipy.optimize` [28]. The comparison of the three pictures shows how different the irreversible solutions (**b** and **c**) are from the reversible one (**a**). The comparison also shows that the p_k law with properly identified p_k parameters provides a rather accurate approximation of the response of the mesoscopic lamination model.

Using the same p_k parameters uniformly for all mesh elements discretizing the laminated core region Ω_core leads however to large errors, as illustrated in Figure 2, because the magnetic fields are quite different from place to place (rotor, stator, teeth, yoke, etc.). Optimized p_k parameters must therefore be determined elementwise in the finite element mesh. Solving the global minimization problem (2) in each element of Ω_core would however be particularly expensive, since each minimization takes up to several seconds. It would also not guarantee any regularity on the distribution of the p_k parameters over the mesh, which can negatively influence the convergence of the nonlinear macroscale problem. An alternative approach is offered by the training of a NN to perform the $\mathbf{H} \mapsto p_k$ mapping efficiently, as explained in the next section.

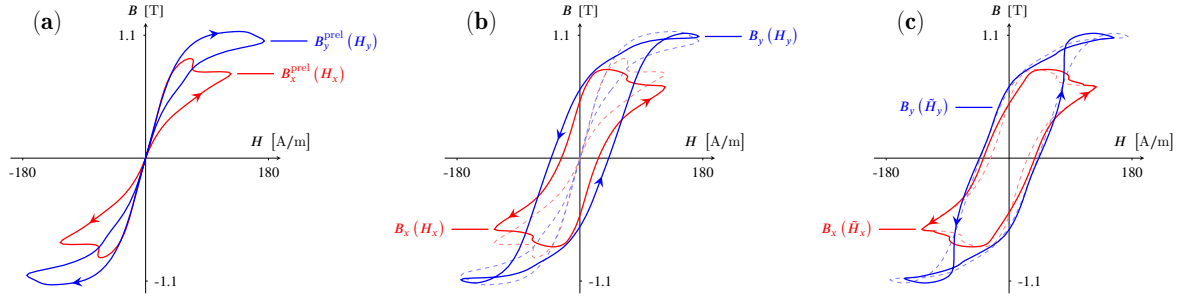

FIGURE 1 (**a**) Cycles $(\mathbf{H}, \mathbf{B}^{\text{prel}})$ obtained from the preliminary simulation with conventional anhysteretic material law. The x and y cycles have opposite orientations and their areas sum up to zero because the anhysteretic material is non-dissipative. (**b**) Cycles $(\mathbf{H}, \mathbf{B})$ with $\mathbf{B}$ obtained from the lamination model (solid lines). The cycles have now the same orientation and the sum of their areas represents the magnetic losses. The cycles of (**a**) are recalled with dashed lines. (**c**) Cycles $(\tilde{\mathbf{H}}, \mathbf{B})$ where $\tilde{\mathbf{H}}$ is obtained by the p_k law, with p_k parameters obtained by solving (2). The cycles of (**b**) are recalled with dashed lines.

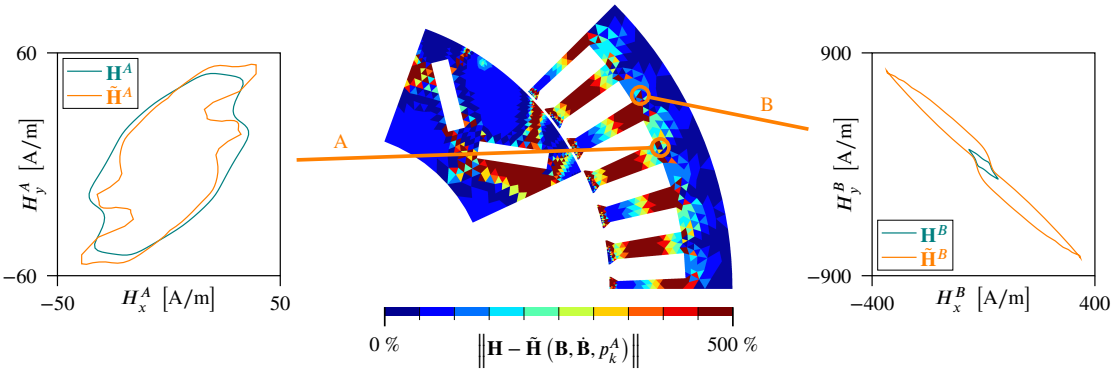

FIGURE 2 The optimized parameters p_k^A determined with the input sequence $\mathbf{H}^A$ provide a good approximation $\tilde{\mathbf{H}}^A\left(\mathbf{B}^A, \dot{\mathbf{B}}^A, p_k^A\right)$ of the sequence $\mathbf{H}^A$. Using the same set of parameters p_k^A in other elements leads to large errors (up to 500 %). For instance, the sequence $\tilde{\mathbf{H}}^B\left(\mathbf{B}^B, \dot{\mathbf{B}}^B, p_k^A\right)$ is a poor approximation of the sequence $\mathbf{H}^B$.

3 | NEURAL NETWORK ARCHITECTURE

In recent years, NNs have been introduced in various domains of physics and have proven their ability to handle complex tasks at a reduced cost. Multi-scale simulations are no exception. In [32] and [17], recurrent NNs are used to account for history-dependent behavior in elasto-plastic materials subjected to random loading paths. In [23], the relative differential permeability of a ferromagnetic steel is predicted by a simple NN-based hysteresis model with the motivation to later use it in 2D FE simulations. In [8], the transformer architecture, originally developed for natural language processing tasks, is adapted to model dynamical systems. The review [2] also presents artificial intelligence as a revolution in multi-scale modeling with impacts on all domains of sciences, including medicine, biology, fluid mechanics, etc.

In this paper, the auto-encoder architecture is adapted so that an encoder block implements the $\mathbf{H} \mapsto p_k$ mapping (2), as illustrated in Figure 3. From a given input $\mathbf{H}$ sequence, augmented with the fundamental frequency f, the encoder block provides values for the p_k parameters. With these values, the p_k law output, $\tilde{\mathbf{H}}\left(\mathbf{B}(\mathbf{H}), \dot{\mathbf{B}}(\mathbf{H}), p_k\right)$, is evaluated by (1), using the response $\mathbf{B}(\mathbf{H})$ of the lamination model to the same sequence $\mathbf{H}$, and its time derivative $\dot{\mathbf{B}}(\mathbf{H})$. The error (3), or mismatch, $\left\|\mathbf{H} - \tilde{\mathbf{H}}\right\|$ is then computed and back-propagated to train the encoder.

The mismatch $\left\|\mathbf{H} - \tilde{\mathbf{H}}\right\|$ between the input and the output of our encoder-decoder architecture is thus evaluated by using the p_k law as decoder. Put another way, the decoder is not learned in our approach, but is replaced by a known function. The main advantage of this is that the training set is composed of $(\mathbf{H}, \mathbf{B})$ pairs, instead of the $(\mathbf{H}, p_k)$ pairs one would need to train the encoder directly. There is therefore no need for the costly generation of exact $(\mathbf{H}, p_k)$ pairs, and the homogenized lamination

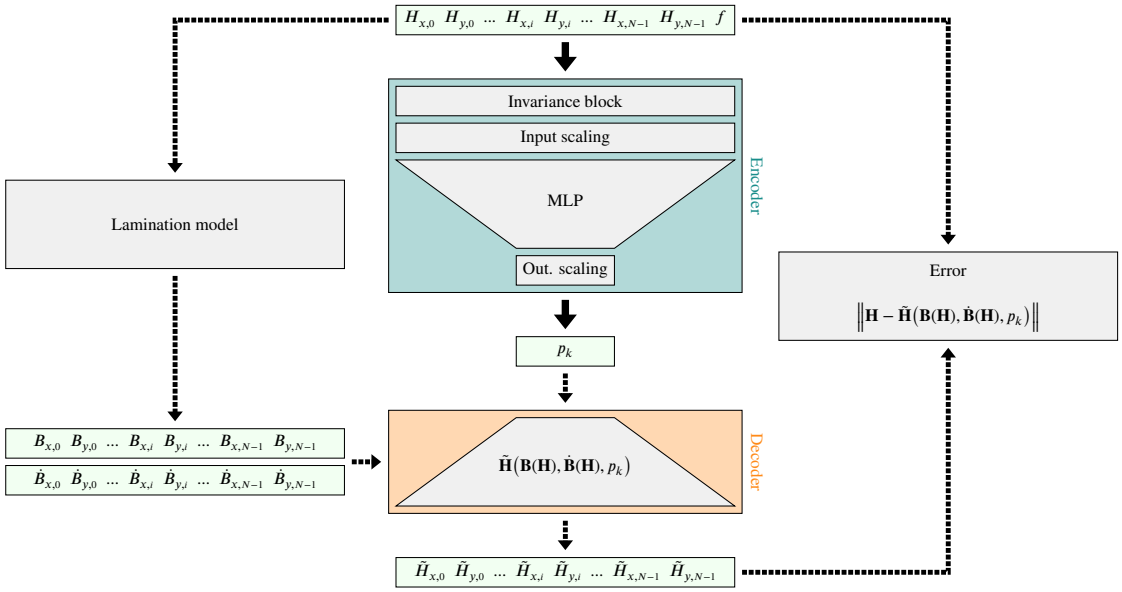

FIGURE 3 NN architecture with auto-encoder-like training. The encoder handles the $\mathbf{H} \mapsto p_k$ mapping (2) while the decoder (the p_k law with the appropriate $\mathbf{B}$ and $\dot{\mathbf{B}}$ sequences) provides an approximation $\tilde{\mathbf{H}}\big(\mathbf{B}(\mathbf{H}), \dot{\mathbf{B}}(\mathbf{H}), p_k\big)$ (1) of the input sequence $\mathbf{H}$. The Multi-Layer Perceptron (MLP) from the encoder is trained by back-propagating the error (3) between the two. Compared to [22], the architecture has been generalized to handle vector magnetic fields, thus allowing it to be used in 2D electrical machine simulations.

model, which evaluates very fast, is sufficient to generate the training set. Once the training is completed, only the encoder is used for the evaluation. It will be further referred to as NN for conciseness.

The NN is composed of four blocks. The first is an invariance block that ensures that equivalent input sequences, i.e., sequences that are image of each other by a rotation and/or a phase shift, are considered identically (see Section 3.1). The next block in the encoder is the input scaling (see Section 3.2). It brings magnetic field values, which can be very large, into a range suited for learning. It is followed by the trainable part of the NN, which is a simple Multi-Layer Perceptron (MLP) with two hidden layers of 700 and 300 neurons, and `ReLU` and `tanh` activation functions, respectively. The output layer has 6 neurons (for the 6 p_k parameters) with linear activation functions. Finally, an output scaling is applied independently to each output, so that each p_k parameter is allowed to vary in its own range (see Section 3.3).

3.1 | Invariance block

Two sequences are equivalent if they are image of each other by a rotation in the $x-y$ plane and/or a phase shift. Due to the high level of symmetry of electrical machines, equivalent sequences are frequently encountered, and the $\mathbf{H} \mapsto p_k$ mapping (2) should return identical p_k parameters for equivalent sequences. To make the mapping (2) rotation-and-phase invariant, the invariance block shifts and rotates the input sequences, so that all equivalent sequences are reduced to a single representative (see Figure 4).

The algorithm of the invariance block first finds the index $\bar{i}$ of the timestep with the highest norm $|\mathbf{H}|$. Rolling the sequence by $-\bar{i}$ places ensures phase invariance. The rotation invariance is obtained by applying a geometric rotation to the sequence in the $x-y$ plane by an angle $\bar{\alpha}$ chosen to ensure that the direction of $\mathbf{H}_{\bar{i}}$ makes an angle of $45°$ with the x–axis, i.e., that the x and y components of $\mathbf{H}_{\bar{i}}$ have identical positive values. The x and y components of the rotated sequence $\mathbf{H}^{\text{inv}}$ are thus better balanced, which facilitates equal weighting when training the NN.

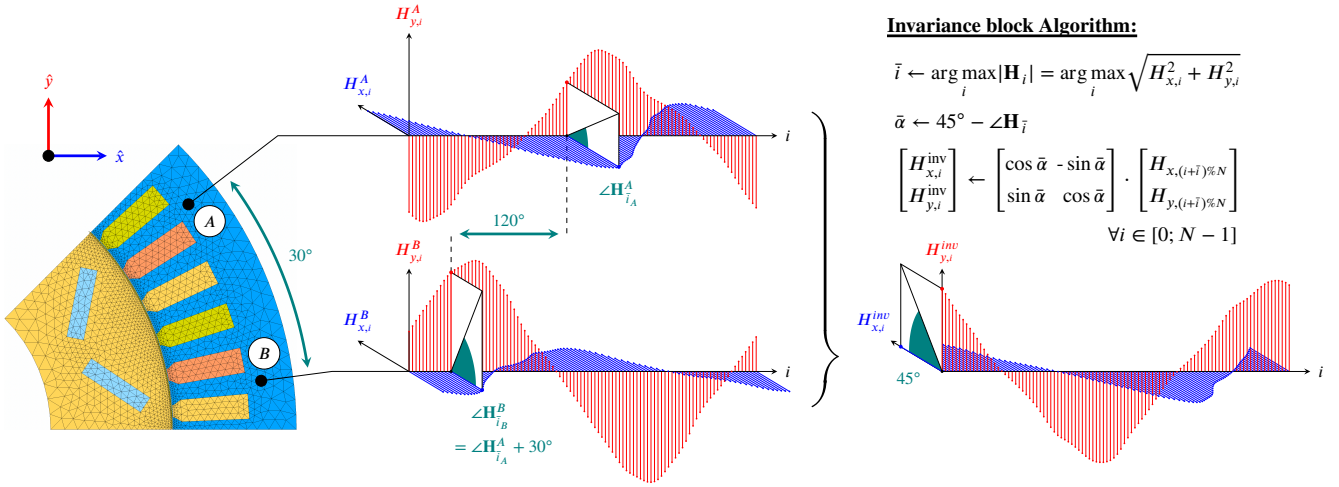

FIGURE 4 The elements indicated A and B see equivalent $\mathbf{H}$ sequences, up to a $120°$ phase shift and a $30°$ rotation. The algorithm of the invariance block applies a shift in time and a rotation so that (i) the timestep with the highest field norm is placed at the beginning of the sequence, and (ii) its x and y components have equal positive values.

3.2 | Input scaling

In the NN literature, data pre-processing is presented as a key concept for facilitating back-propagation [19, 26, 15]. In particular, all inputs should be scaled to fit within similar ranges, typically close to unity. Furthermore, for the input scaling to be relevant, it is important to understand what composes the inputs. In this paper, the focus is on $\mathbf{H}$ sequences encountered in ferromagnetic materials. These materials exhibit saturation, meaning that, while the magnetic flux density $\mathbf{B}$ grows rapidly (and almost linearly) with the magnetic field $\mathbf{H}$ for small values of $\mathbf{H}$, for large values of $\mathbf{H}$ it only grows very slowly. Therefore, in order to reduce the sensitivity of the NN to the very large values of the fields in the saturated range, the input scaling uses a hyperbolic tangent term in addition to a linear term

$$\mathbf{H}^{\text{scaled}}(\mathbf{H}) = \frac{\mathbf{H}}{|\mathbf{H}|_{\max}} + \tanh\left(\frac{|\mathbf{H}|}{2\,H_{\text{knee}}}\right)\frac{\mathbf{H}}{|\mathbf{H}|} \tag{4}$$

where $|\mathbf{H}|_{\max}$ is the maximum norm of $\mathbf{H}$ in the whole training dataset, and H_{knee} is the norm of $\mathbf{H}$ at the saturation knee i.e. where $\mathbf{B}$ stops evolving linearly with $\mathbf{H}$. For the ferromagnetic sheets of the application at hand (see Section 4), H_{knee} is about $1\,000$ A/m. The effect of this scaling is illustrated in Figure 5.

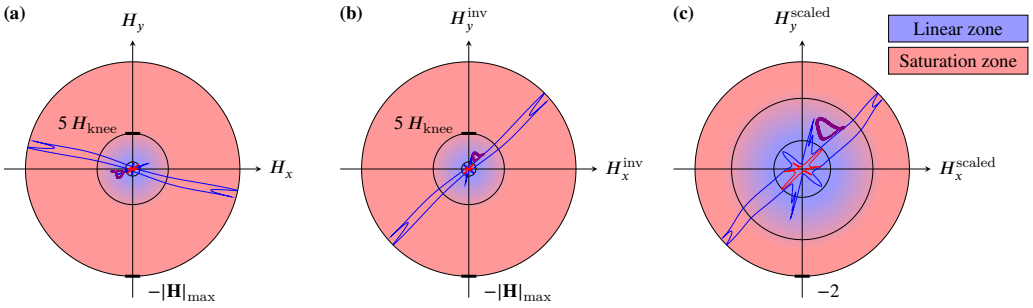

FIGURE 5 (**a**) H_y (H_x) curves for a dataset of three $\mathbf{H}$ sequences. The smaller circle has a radius H_{knee} and represents the boundary after which the material starts saturating. At a radius of $5\,H_{\text{knee}}$, the material is considered to be fully saturated ($\tanh(5/2) \approx 0.99$). (**b**) Curves obtained after running the algorithm of the invariance block. The curves are now aligned with a slope of $45°$. (**c**) Final curves given at the entrance of the MLP, after running the algorithm of the invariance block and applying the proposed scaling. No information is lost for high fields (saturation zone) and it is now easy to see the behavior at low fields (linear zone).

3.3 | Output scaling

In Section 2, the p_k law (1) was introduced together with physical interpretations of the p_k parameters. Based on these interpretations, each p_k parameter is assigned an *a priori* estimate, denoted as $\bar{p}_k$. Table 1 lists these estimated values along with their respective lower bounds $p_{k_{\min}}$. Indeed, it is important to note that all p_k parameters are defined as positive and must therefore be greater than or equal to zero. In addition, $p_{2_{\min}}$ is set to 1 to avoid any division by zero in the derivative and $p_{5_{\min}}$ is set to a small positive value ϵ, again, to avoid any division by zero.

TABLE 1 A-priori estimates $\bar{p}_k$ of the p_k parameters.

$\bar{p}_0$	$\bar{p}_1$	$\bar{p}_2$	$\bar{p}_3$	$\bar{p}_4$	$\bar{p}_5$	$p_{0_{\min}}$	$p_{1_{\min}}$	$p_{2_{\min}}$	$p_{3_{\min}}$	$p_{4_{\min}}$	$p_{5_{\min}}$
μ^{-1}	${B_{\text{knee}}}^{-2}$	5	$\sigma\, d^2\,/\,12$	H_c	$\omega\, B_{\text{sat}}$	0	0	1	0	0	ϵ

From Table 1, it is evident that the p_k parameters can vary across significantly different ranges. This variation may hinder the training process if appropriate output scaling is not implemented. Without such scaling, parameters with smaller ranges will indeed disproportionately influence the loss function compared to those with larger ranges. The gradient updates for the smaller-range parameters would thus be larger, leading to a steeper descent in the error landscape. Smaller-range parameters may therefore oscillate around some minima while larger-range parameters may only leisurely step to a minimum.

To balance the training process and ensure lower bounds at $p_{k_{\min}}$, the k outputs o_k of the MLP are transformed as

$$p_k = \texttt{softplus}\left(o_k - 1 - \frac{p_{k_{\min}}}{\bar{p}_k}\right)\bar{p}_k + p_{k_{\min}} \tag{5}$$

where $\texttt{softplus}\,(x) = \ln\left(1 + e^x\right)$ is a smooth approximation of the `ReLU` function. Figure 6 illustrates the shape of (5), providing a visual basis for assessing its benefits. At first, it ensures the p_k parameters are always higher than the specified minimum values $p_{k_{\min}}$. Secondly, it scales large o_k outputs with the *a priori* estimates $\bar{p}_k$, hence mitigating the differences in range while maintaining good sensibility for small o_k values. Finally, it offers an effective starting point for the NN training. Indeed, since the MLP weights are initialized with small random values around zero, the MLP outputs o_k are expected close to zero at the beginning of the training. This translates to initial p_k values close to the *a priori* estimates $\bar{p}_k$.

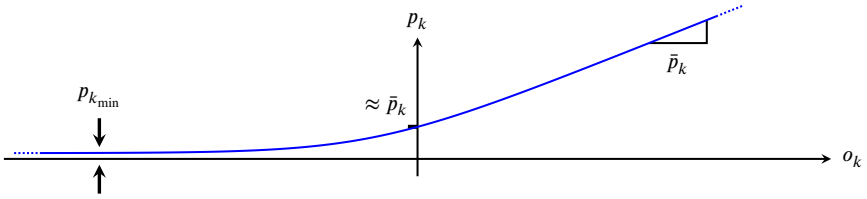

FIGURE 6 Typical shape of the output scaling function (5).

4 | APPLICATION AND RESULTS

The considered application example is a two pole-pair 220 kW switched reluctance motor (see Figure 7). The motor is supplied with 3-phase sinusoidal currents of 217.31 A_{RMS} at a working frequency of 65 Hz, which corresponds to a nominal speed of 1950 rpm. Stator and rotor are built as stacks of M235-35A steel laminations. The flux barrier in the rotor are the results of a shape optimization to minimize the torque ripple [4]. The model is discretized with circa 15 000 nodes and 26 000 triangles using Gmsh [9].

The complete workflow of the proposed technique to run a FE simulation of this motor with core losses using the p_k law described above is given in Figure 8.

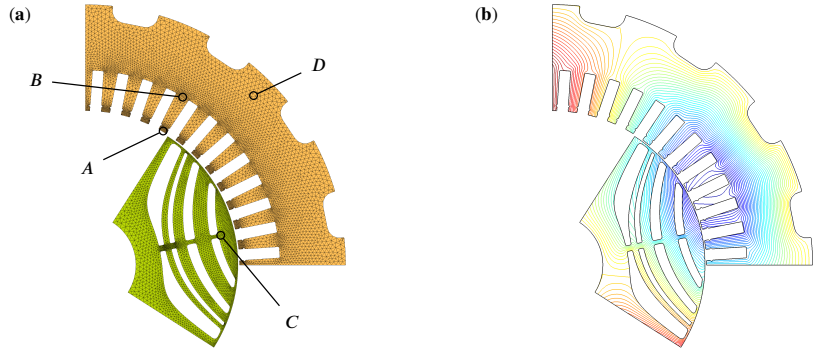

FIGURE 7 Mesh (**a**) and field lines (**b**) in one pole of the analyzed switched reluctance motor.

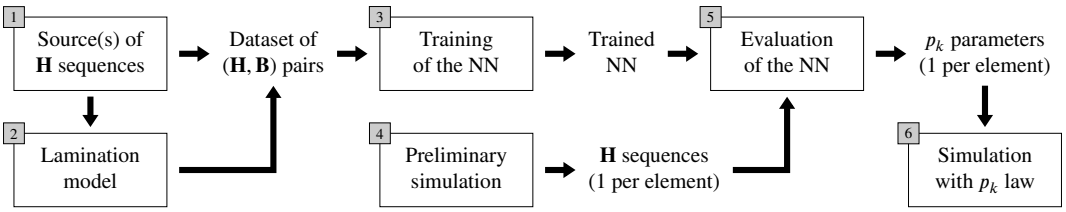

FIGURE 8 Complete workflow of the proposed method.

4.1 | Preliminary simulation

To complete the workflow presented in Figure 8, it is necessary to run a preliminary 2D FE simulation of the motor. This is here handled with a conventional anhysteretic saturation curve. The weak formulation is as follows: find the magnetic vector potential $\mathbf{A}$ on the domain of analysis Ω such that

$$\left(\mu_0^{-1}\,\mathrm{curl}\,\mathbf{A}, \mathrm{curl}\,\mathbf{A}'\right)_{\Omega\setminus\Omega_{\mathrm{core}}} + \left(\mathbf{H}^{\mathrm{anhys}}(\mathrm{curl}\,\mathbf{A}), \mathrm{curl}\,\mathbf{A}'\right)_{\Omega_{\mathrm{core}}} - \left(\mathbf{j}_s, \mathbf{A}'\right)_{\Omega_s} = 0 \tag{6}$$

holds for all test function $\mathbf{A}'$ in the appropriate function space. Homogeneous Dirichlet boundary conditions are applied on the zero flux inner and outer surfaces of Ω. Periodicity boundary conditions are applied on the radial lateral faces, as well as in the air gap, so as to only simulate one pole of the machine. The magnetic vector potential field $\mathbf{A}$ is discretized with first order Lagrange elements using the open source FE solver GetDP [6]. The anhysteretic saturation curve $\mathbf{H}^{\mathrm{anhys}}(\mathbf{B})$, with $\mathbf{B} = \mathrm{curl}\,\mathbf{A}$, is extracted from Epstein measurements performed on M235-35A steel laminations. A sinusoidal 3-phase current density $\mathbf{j}_s$, calculated from the stator current and the number of windings, is injected in the stator stranded conductor region Ω_s. Formulation (6) is purely static, as no eddy currents are modeled in the reluctance motor when the anhysteretic law is used to model the laminated core Ω_{core} and currents are imposed in the stator windings. A single period is thus simulated, using $N = 600$ steps to reach a sufficient accuracy for the purposes of our analysis. Note that in other applications where eddy currents outside of the laminated core need to be considered, a standard magneto-quasistatic formulation would be used instead [20], and the simulation should be carried out over a few periods until reaching steady-state.

4.2 | Training and evaluation of the NN

The training set consists of a large number of $(\mathbf{H}, \mathbf{B})$ pairs of sequences. The Box 1 in Figure 8, labeled "Source(s) of **H** sequences", stands for any process which generates **H** sequences relevant for the training of the NN. These can be sequences obtained by FE simulations at different operating points, experimental data, or randomly generated sequences as was done in [22]. All the **H** sequences gathered in Box 1 are then injected as excitations into the mesoscopic lamination model to obtain the lamination response $\mathbf{B}(\mathbf{H})$ and therefore complete the training pairs $(\mathbf{H}, \mathbf{B})$.

In the simplest option, the $\mathbf{H}$ sequences used to train the NN are those obtained from the preliminary simulation of Section 4.1. We call *purely specific* this approach, for which the sequences used for training are also the ones that will be used later at the evaluation step (Box 5 in Figure 8). Note that the usual distinction between training, validation and test sets is therefore not relevant in this case.

The training of the NN is then performed with the generated training set using the PyTorch framework [21]. We have used a learning rate of 10^{-4}, batches of 256 $(\mathbf{H}, \mathbf{B})$ pairs and the `RAdam` optimizer. The training process performs $50\,000$ updates of the NN, the mean error being evaluated every $1\,000$ updates. The final NN is the one that has led to the lowest error across all evaluations. In total, solving the mesoscopic lamination model for all the sequences of Ω_{core} (17 000 sequences) originating from the preliminary simulation (see Section 4.1) takes about 10 minutes on a standard laptop computer while the training lasts about 20 minutes.

Once the training is completed, the performance of the trained NN is assessed by visualizing the mismatch between the mesoscopic model $(\mathbf{H}, \mathbf{B})$ and its restitution $(\tilde{\mathbf{H}}, \mathbf{B})$ by the p_k law (1) with the p_k parameters now obtained from the trained NN, with the sequence $\mathbf{H}$ as input. This is first done at the four characteristic points, A, B, C and D, indicated in Figure 7. The first column in Figure 9 compares the sequences $\mathbf{H}$ and $\tilde{\mathbf{H}}$ over one period of time. The second column compares the loci in the $x-y$ plane of $\mathbf{H}$ and $\tilde{\mathbf{H}}$. Finally, the third column displays four **HB** cycles: (H_x, B_x) and (H_y, B_y) on one hand, and $(\tilde{H}_x, B_x)$ and $(\tilde{H}_y, B_y)$ on the other hand. One observes that the agreement is globally excellent, despite the complexity and diversity of the magnetic fields at the different locations examined in the machine. The accuracy of the method is examined more thoroughly in the next section.

4.3 | Impact of training on accuracy

The mismatch $\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|$ (3) has two origins. It is for one part due to the p_k law itself. Indeed, with only six parameters, the p_k law (1) cannot match exactly the response of the complex mesoscopic lamination model. There exists thus an irreducible error entailed by the fact that the complex phenomena happening inside the laminations are necessarily expressed in the FE model through a parametric law, the p_k law, whose algebraic expression is fixed. The second component of the mismatch $\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|$ is the approximation introduced by the NN itself. This second component may vary from one learning to the other since the initial weights of the MLP are set randomly, and each update to these weights during training is based on a randomly selected batch of the training data. Running multiple times the training process thus produces distinct NNs, which may return different p_k values to the same input data $\mathbf{H}$. In order to distinguish the irreducible error due to the p_k law from the error due to the values of the NN-determined parameters, one defines

$$\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|_{\text{irred}} = \min_{\text{NN}} \left\|\mathbf{H}-\tilde{\mathbf{H}}\right\| . \tag{7}$$

This irreducible error can be estimated by evaluating the $\mathbf{H} \mapsto p_k$ mapping (2) with a reference global optimizer. As in Section 2, we used the global dual annealing (GDA) optimizer from `scipy.optimize`, which is expected to reach the true global minimum of the error function (3), independently of any NN consideration.

The p_k parameters and the corresponding mismatch $\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|$ have been evaluated in all elements of Ω_{core}, and are displayed in Figure 10. The upper row has been obtained with the GDA optimizer, whereas the last two rows have been obtained with two NNs (NN_1 and NN_2) trained identically on the same training set.

The mismatch amounts to 5.8 %, 6.9 % and 7 % in average, for GDA, NN_1 and NN_2 respectively. As expected, the lowest mismatch is reached by the GDA optimizer, but the corresponding spatial distribution of the p_k parameter values is rather jagged and irregular (see, e.g., p_5 in the upper row). The ranges of p_k values obtained over the whole machine with the GDA optimizer are also significantly larger than those of the smoother distributions obtained with the NNs. This is due to the fact that the GDA optimizer processes each sequence independently of any other, whereas the NN training minimizes the mismatch $\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|$ (3) in average over the whole training set with a limited number of weights and biases. It results into considerably more regular distributions of the p_k parameters.

Different training runs may however output slightly different p_k parameters, as can already be seen by comparison of the last two rows in Figure 10. The variability introduced by the training can be further analyzed by comparing the $\tilde{\mathbf{H}}$ loci obtained from six different training runs (the two of Figure 10 plus four additional instances noted NN_3 to NN_6) with the locus obtained with the GDA optimizer. This is done in the Figures 11 and 12 for two particular $\mathbf{H}$ sequences.

The left parts of the figures display the different sets of p_k parameters in the same ranges as those of Figure 10. For readability, the p_k values and the mismatch corresponding to a same set (same NN_i or GDA) are linked in the figure with a smooth line

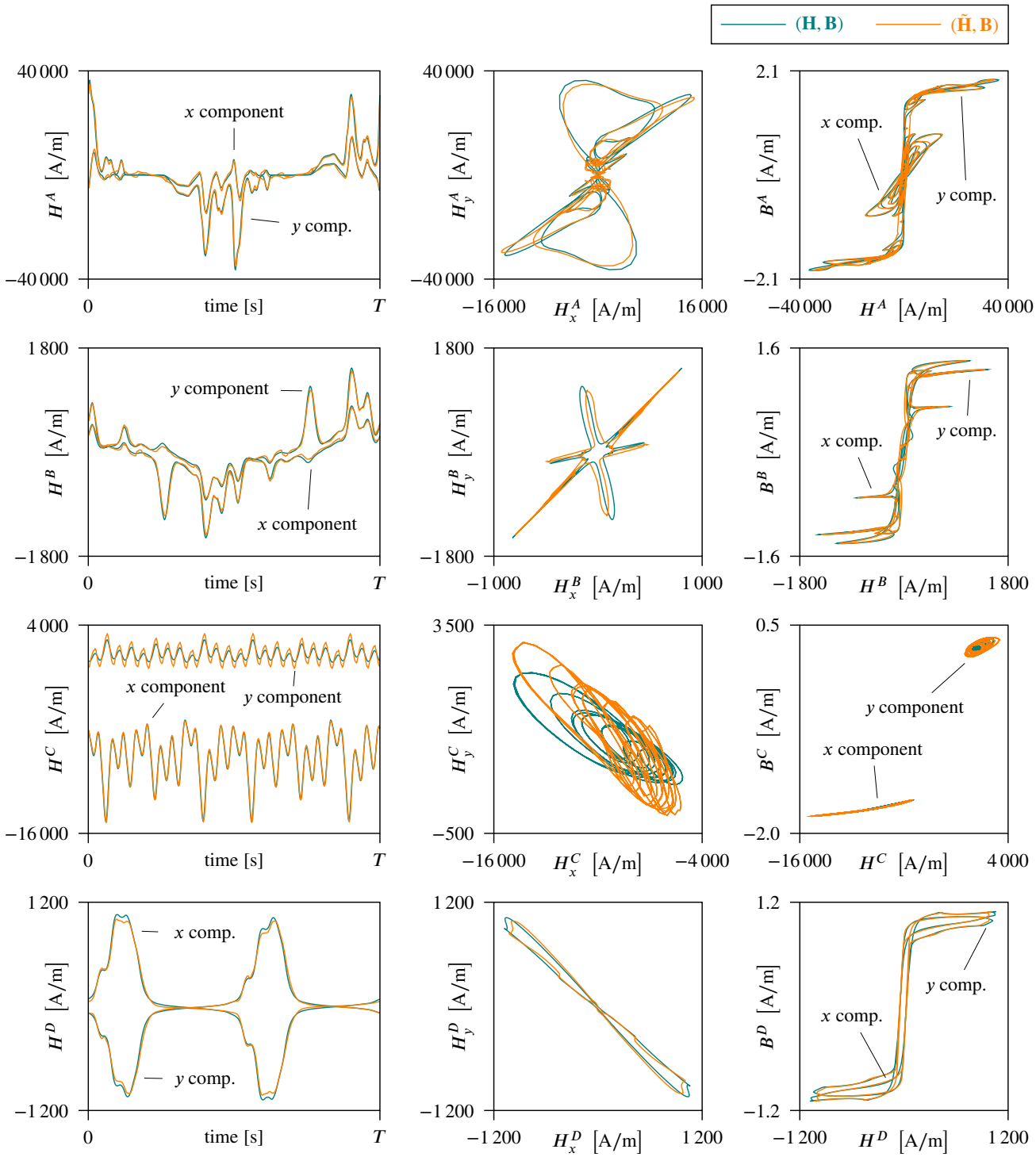

FIGURE 9 Comparison at different locations in the motor (see Figure 7) of the mesoscopic model with its restitution by the p_k law.

with no physical meaning. The mismatches are similar for the six generated NNs (solid lines). They all lay between 11 % and 11.4 % for Figure 11, and between 5.7 % and 6.9 % for Figure 12. The GDA optimizer (dotted black line) yields again a lower mismatch $\|\mathbf{H} - \tilde{\mathbf{H}}\|$ (3), which is however only slightly lower. This indicates that the major part of the mismatch $\|\mathbf{H} - \tilde{\mathbf{H}}\|$ is the irreducible error (7) that we have defined as the error granted by the GDA optimizer. This irreducible error amounts to 10.94 % in Figure 11, to 5.28 % in Figure 12, and to 5.8 % in average over the whole machine.

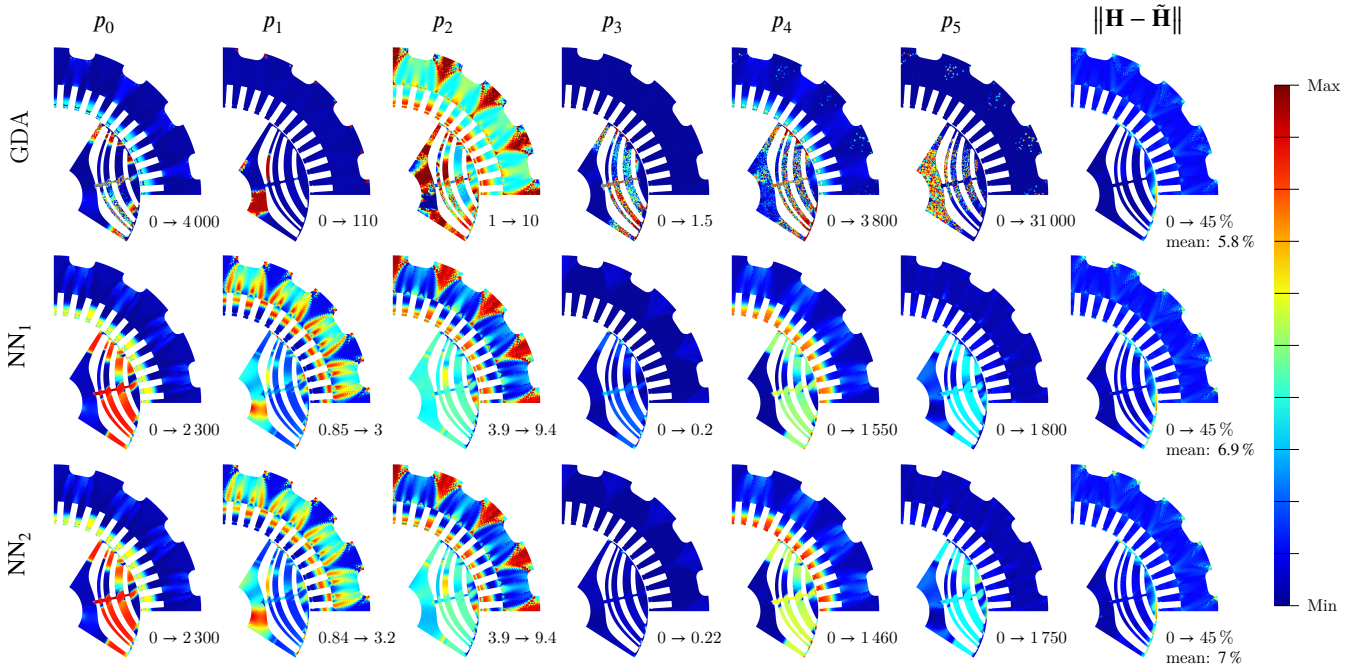

FIGURE 10 The first six columns present the spatial distribution of parameters p_0 to p_5, the last one presents the distribution of the corresponding mismatch $\left\|\mathbf{H}-\tilde{\mathbf{H}}\right\|$ (3). The first line is obtained after running the GDA optimizer over all the $\mathbf{H}$ sequences of the dataset. The second and third lines are obtained from the evaluation of two identically trained NN instances (NN$_1$ & NN$_2$).

The additional error due to the training is much smaller. It is below 0.5 % in Figure 11 and below 1.7 % in Figure 12. This observation is confirmed by the $\mathbf{H}$ loci presented in the right part of the Figures 11 and 12. The six NN loci (solid lines) and the GDA locus (dotted black line), which are all calculated with the p_k law, form a bunch of curves very close to each other, whereas the reference $\mathbf{H}$ locus (red dotted line) deviates significantly from this bunch, due to the somewhat larger irreducible error. A properly trained NN proves therefore able to successfully perform the regression work needed to identify close to optimal p_k parameters, introducing only an additional mismatch of less than 2 % in average to the irreducible error due to the fixed algebraic form of the p_k law.

One finally notes that the computed p_k parameters are quite different from the $\bar{p}_k$ values (indicated with dots in Figures 11 and 12) that were guessed on basis of physical considerations. This means that machine learning is able to provide much better solutions than basic physical intuition.

4.4 | Simulation with the $\mathbf{p_k}$ law

Once the six p_k parameters are known for each element in the laminated core region Ω_{core}, the FE computation can be restarted, with this time the irreversible p_k law (1) as material law instead of the reversible anhysteretic law, i.e., with the formulation

$$\left(\mu_0^{-1}\operatorname{curl}\mathbf{A},\operatorname{curl}\mathbf{A}'\right)_{\Omega\setminus\Omega_{\text{core}}}+\left(\tilde{\mathbf{H}}(\mathbf{B},\dot{\mathbf{B}},p_k),\operatorname{curl}\mathbf{A}'\right)_{\Omega_{\text{core}}}-\left(\mathbf{j}_s,\mathbf{A}'\right)_{\Omega_s}=0 \tag{8}$$

where the derivatives $\dot{\mathbf{B}}$ are discretized with the implicit Euler scheme.

It is useful at this stage to take a quick look back at Figure 1, in order to clearly see where we are. The cycles (**a**) on the left-hand side in that figure stand for the solution that would be obtained with the anhysteretic model. On the other hand, the lamination model, our ground truth, tells us that reality is close to the cycles (**b**) presented at the center of the same figure. From Section 4.2, one also knows at this point that the p_k law properly supplied with NN-identified parameters produces a solution close to the cycles (**c**). The latter deviates from the target cycles (**b**) by a relative error below 7 % in average (due to the p_k law).

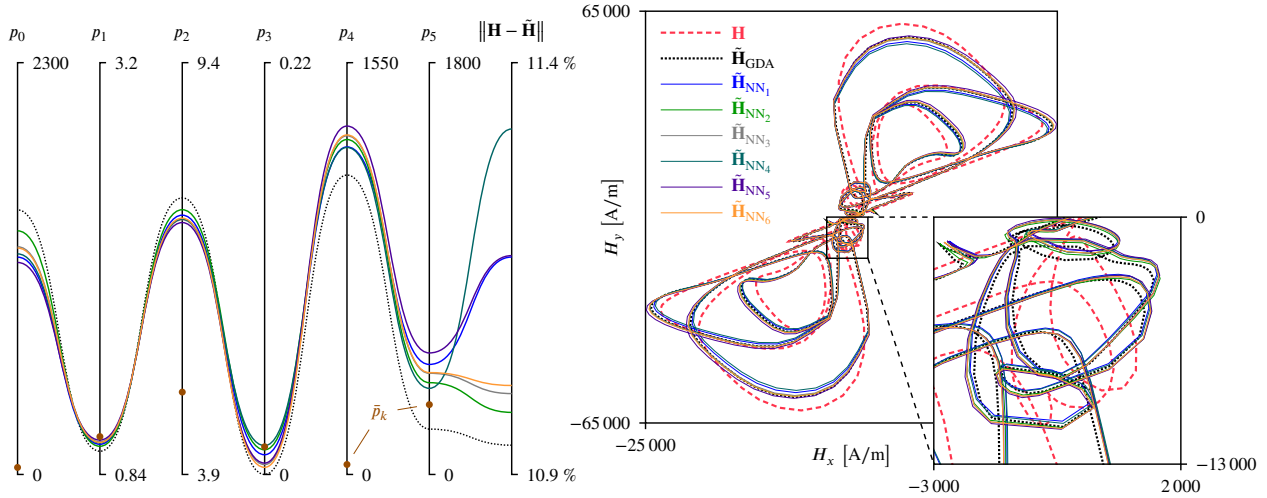

FIGURE 11 Considering the sequence **H** obtained at point *A* from Figure 7, comparison of the p_k sets, mismatches and magnetic field loci obtained with the GDA optimizer and six different NN instances, trained with identical dataset and hyper-parameters. For readability, the p_k values and the mismatch corresponding to a same set (same NN_i or GDA) are linked with a smooth line with no physical meaning.

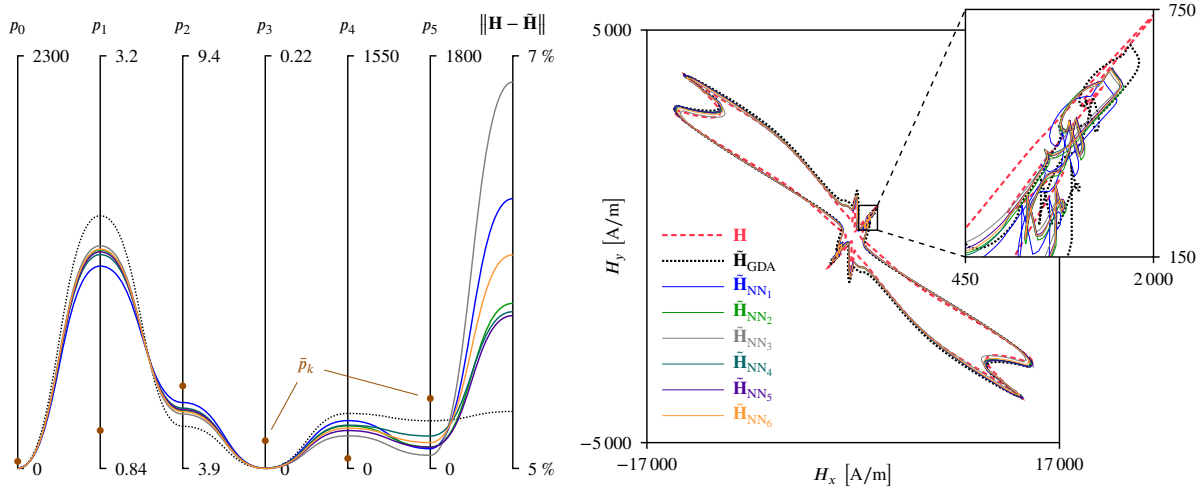

FIGURE 12 Same comparison as for Figure 11, performed at another point in the motor.

However, despite this irreducible error, Figure 1 clearly shows that the proposed method offers a solution much closer to reality than the conventional anhysteretic one.

The method proposed in this paper gives insight into the details of the magnetic loss distribution in the electrical machine, as shown in Figure 13, both in linear and logarithmic scale. But there is more than just loss computation. Accounting for magnetic losses in the simulation has also, for instance, an impact on the computed torque, which is reduced in average by an amount equal to the total magnetic losses divided by the angular velocity. This torque reduction is expected to be about 5 Nm per kW of magnetic losses for this motor, which is consistent with what can be seen in Figure 14. Magnetic losses are also represented on the right-hand side in the same figure. They are computed by integration on Ω_{core} of the magnetic loss density given by

$$\dot{W}_{\text{mag}}\left(\dot{\mathbf{B}}, p_k\right) = p_3|\dot{\mathbf{B}}|^2 + \frac{p_4}{\sqrt{p_5^2 + |\dot{\mathbf{B}}|^2}}|\dot{\mathbf{B}}|^2. \tag{9}$$

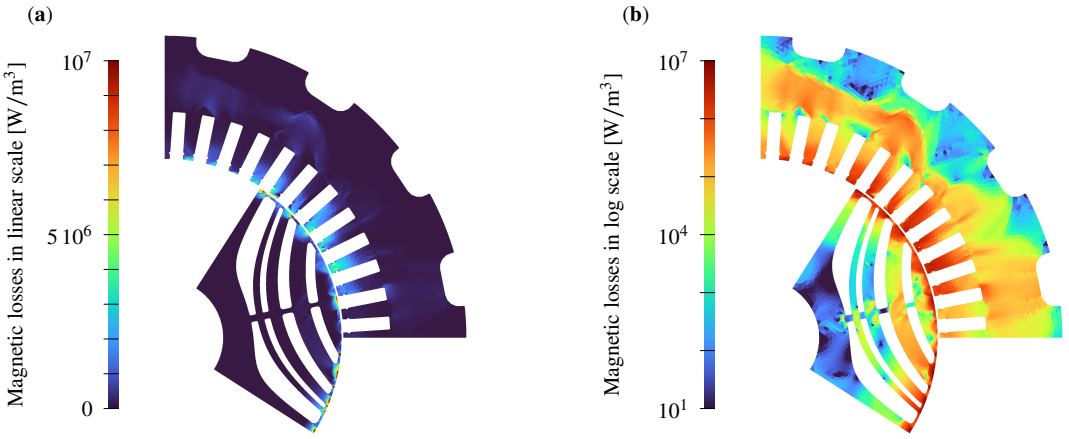

FIGURE 13 Distribution of magnetic losses across Ω_{core} at a particular instant of time. **(a)** linear scale, **(b)** logarithmic scale.

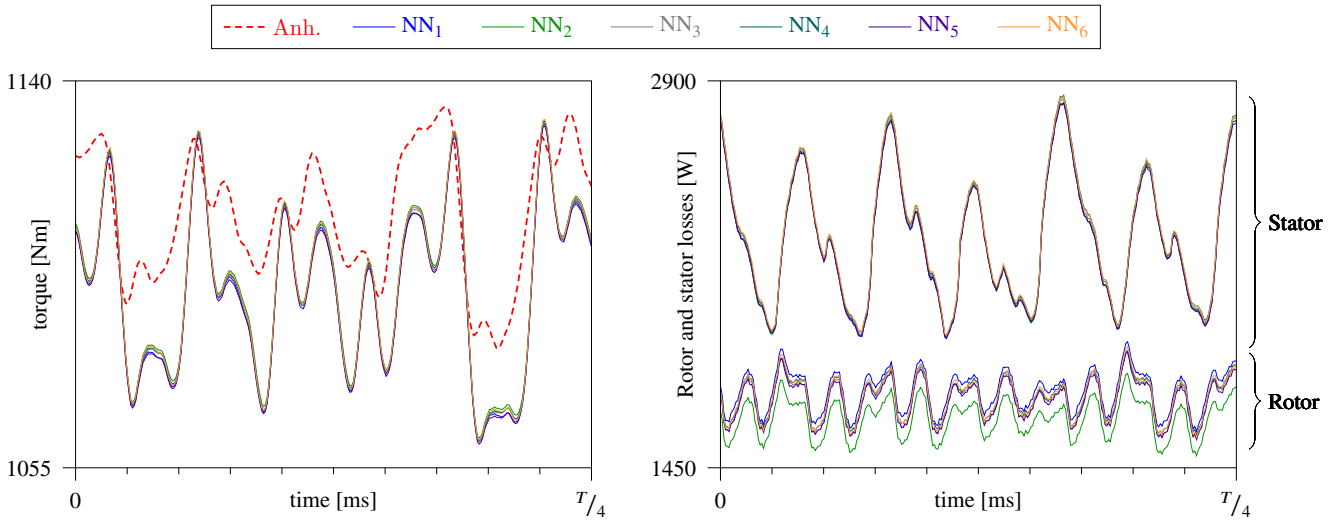

FIGURE 14 Left: comparison of the torque curves obtained with the anhysteretic model and with the p_k law, using the six NNs (NN_1 to NN_6 presented in Section 4.3). Right: comparison of the magnetic loss curves at the stator and at the rotor computed with the p_k law using the six aforementioned NNs.

The computed torque appears to be practically insensitive to the variability introduced in the modeling by the NN-training, since the six computed curves, NN_1 to NN_6, are nearly identical. This also holds true for the losses at the stator, for which the curves over time are nearly superposing. At the rotor side, the variability due to training is however more marked. In fact, when analyzing the accuracy of the computed magnetic losses, a difference should be made between rotor and stator. The harmonic content of the fields at the rotor side in a synchronous machine is quite different from that of the fields at the stator side. The steady rotation indeed wipes off the fundamental field harmonic at the rotor side, leaving there only a DC component and higher harmonics. In comparison to stator sequences, this adds two difficulties for the NN. At first, the dominant harmonic at the rotor side, which is here the $12f$ harmonic (See Figure 9, third row), is more poorly discretized with 600 time steps per period than the stator field, for which the fundamental harmonic of frequency f is dominant. Secondly, the DC shift places the rotor $\mathbf{H}$ sequences in a region where the input scaling (4) is less sensitive.

4.5 | Computational complexity

In order to show the efficiency of the proposed approach, one can now provide a brief computational complexity discussion. The method can be split into 3 main steps: (i) a preliminary FE simulation with an approximate (anhysteretic) material law in Ω_{core} to generate relevant training sequences $\mathbf{H}$; (ii) the training of the NN and its evaluation to obtain the p_k parameter values in all elements of Ω_{core}; (iii) a second FE simulation with the p_k law. The steps (ii) and (iii) represent the overhead to be paid to benefit from the more accurate lamination model. If N_{core} denotes the number of finite elements in Ω_{core}, step (ii) requires N_{core} solvings of the mesoscale model (about 10 minutes with $N_{\text{core}} = 17\,000$), 50 000 backpropagations (about 20 minutes), and N_{core} evaluations of the final NN (around 10 seconds). The computational time of step (iii) depends on the number of nonlinear Newton-Raphson iterations needed to reach a sufficient accuracy. The nonlinear p_k law (1) has a Jacobian that can be evaluated and assembled exactly (see Appendix B). This ensures monotonic convergence of the Newton-Raphson scheme at each time step, as illustrated in Figure 15. Compared to the anhysteretic simulation, one however observes that the number of iterations is larger and that the rate of convergence significantly degrades once the relative residual reaches below 10^{-4}. This is due to the fact that, because of the elementwise character of the p_k maps, neighboring elements have to obey material laws with different parameter values. Our experience shows that step (iii) costs in practice approximately 3.5 times as much as step (i) using a threshold at 10^{-4} (1 hour and 25 minutes compared to 24 minutes) and approximately 11.5 times as much as step (i) using a threshold at 10^{-5} (4 hours and 40 minutes). The torque and loss curves remaining however unchanged when opting for one or the other threshold. All put together, one could perform on a standard laptop computer a simulation with losses for only around 5.5 times the cost of an anhystertic simulation.

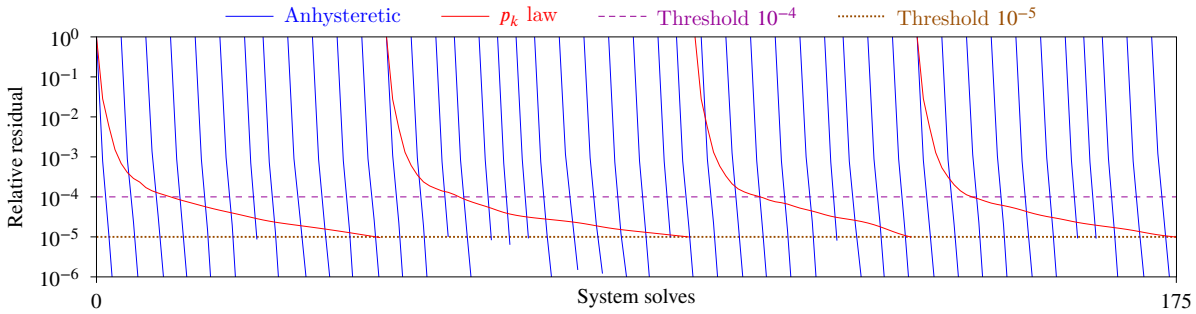

FIGURE 15 Typical convergence history of the non-linear Newton-Raphson iterations, with the anhysteretic material law (blue) and with the p_k law (red). To reach a relative residual of 10^{-4}, 4 iterations are required with the anhysteretic material law in average while the p_k law requires in average 8 iterations. For a threshold at 10^{-5}, it reaches 30.5 iterations in average.

4.6 | Discussion

The approach presented in this paper has several benefits. First, the proposed methodology avoids a direct multiscale coupling by identifying the parameters of a surrogate material law, the multiscale calculations being deferred to the training of the NN. In case of direct multiscale coupling (e.g., HMM [31, 20]), the macroscale and the mesoscale model have to be iterated in time synchronously. A set of hysteresis parameters must also be stored in memory for each element in Ω_{core}, and at each timestep three nonlinear resolutions of the mesoscale model must be carried out, as the macroscale Jacobian matrix must be evaluated by finite differences. Convergence issues are to be expected, since one cannot benefit from the guaranteed monotonic convergence offered by the surrogate p_k law, and the number of necessary nonlinear iterations is unknown. The direct multiscale machinery also implies quite intrusive changes in an existing FE code, whereas the surrogate p_k law can be implemented quite easily in FE codes. Furthermore, even though the lamination model is driven by the magnetic field, the surrogate p_k law is designed as a material law for a conventional magnetic vector potential FE formulation, avoiding the cumbersome inversion of the energy-based hysteresis model [14, 13].

Second, the approach also benefits from the stochastic character of the NN identification, which minimizes the error in average over batches of sequences. In contrast, the GDA optimizer identifies the p_k parameters for one finite element at a time. One has observed that GDA reaches an error below that of the NN in about 2 seconds per sequence (about 9 hours for N_{core} sequences). However, when applied to all finite elements in Ω_{core}, the obtained distribution of p_k values is very irregular. This dramatically hampers convergence, and indicates also that the mismatch is an objective function with many local minima. On the other hand, the NN optimizer grants a slightly larger mismatch, but offers more regularity in the p_k values, which is favorable to convergence and physically relevant. The averaged character of the NN optimizer thus has the added value of making automatically a useful trade-off between mismatch minimization and regularity, whereas the GDA optimizer, by focusing exclusively on the mismatch, delivers eventually p_k distributions that are little usable in practice.

Finally, the trained NN contains knowledge than can be reused for other simulations or for transfer learning, whereas the direct multiscale coupling and the GDA optimizer learn nothing. While this advantage has not been exploited in this paper, it offers clear potential benefits for the application of the technique when variants of a same design have to be analyzed such as in an optimization loop.

5 | CONCLUSION

This paper presents a method to accurately compute magnetic losses in laminated cores by means of 2D finite element simulations. To this aim, an irreversible parametric material law is introduced as a surrogate model in the macroscale simulation. The six parameters of the law are determined elementwise by a neural network so as to best fit the homogenized response of an accurate lamination model. At the end of the day, the presented approach offers a robust model, and an unprecedented insight into the distribution of magnetic losses as well as their impact on the machine performance, at a computational cost comparable to that of a conventional nonlinear finite element simulation. With an appropriate training, it has been demonstrated that the neural network is able to provide close-to-optimum parameter values with an appropriate level of regularity. In contrast to direct multiscale approaches, which imply deep changes in an existing finite element code, the proposed approach defers most of the work to external tools (the lamination model as a stand-alone C code, and PyTorch scripts), which makes it rather easy to implement. The method has been presented in this paper in case of a periodic regime, but it could be applied similarly for a transient computation. In the future, we aim at replacing the fixed algebraic form of the surrogate model by a more versatile NN-trained direct approximation of the $\mathbf{H}$ – $\mathbf{B}$ relationship.

ACKNOWLEDGMENTS

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

[1] S. Q. Antonio, F. R. Fulginei, H. Rimal, and A. Ghanim. On the use of feedforward neural networks to simulate magnetic hysteresis in electrical steels. In *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, pages 119–124. IEEE, 2020.

[2] D. Bishara, Y. Xie, W. Liu, and S. Li. A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials. *Archives of Computational Methods in Engineering*, 30, 08 2022.

[3] E. Cardelli, A. Faba, A. Laudani, S. Quondam Antonio, and A. M. Ghanim. Comparison between different models of magnetic hysteresis in the solution of the team 32 problem. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, page e3103, 2023.

[4] C. De Greef, V. Kluyskens, F. Henrotte, C. Versele, C. Geuzaine, and B. Dehez. Time-efficient multi-physics optimization approaches for the design of synchronous reluctance motors. In *2021 IEEE Energy Conversion Congress and Exposition, ECCE 2021 - Proceedings*. STMicroelectronics, Institute of Electrical and Electronics Engineers Inc., 2021.

[5] E. Dlala. Comparison of models for estimating magnetic core losses in electrical machines using the finite-element method. *IEEE Transactions on Magnetics*, 45(2):716–725, 2009.

[6] P. Dular and C. Geuzaine. GetDP reference manual: the documentation for GetDP, a general environment for the treatment of discrete problems. http://getdp.info.

[7] S. Elfgen, P. Rasilo, and K. Hameyer. Hysteresis and eddy-current losses in electrical steel utilising edge degradation due to cutting effects. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 33(5):e2781, 2020.

[8] N. Geneva and N. Zabaras. Transformers for modeling physical systems. *Neural Networks*, 146:272–289, feb 2022.

[9] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309 – 1331, 09 2009.

[10] J. Gyselinck, L. Vandevelde, J. Melkebeek, P. Dular, F. Henrotte, and W. Legros. Calculation of eddy currents and associated losses in electrical steel laminations. *IEEE Transactions on Magnetics*, 35(3):1191–1194, 1999.

[11] F. Henrotte, S. Steentjes, K. Hameyer, and C. Geuzaine. Iron loss calculation in steel laminations at high frequencies. *IEEE Transactions on Magnetics*, 50(2):333–336, 2014.

[12] F. Henrotte, S. Steentjes, K. Hameyer, and C. Geuzaine. Pragmatic two-step homogenisation technique for ferromagnetic laminated cores. *IET Science, Measurement & Technology*, 9(2):152–159, 2015.

[13] K. Jacques. *Energy-Based Magnetic Hysteresis Models - Theoretical Development and Finite Element Formulations*. PhD thesis, ULiège - Université de Liège, 21 November 2018.

[14] K. Jacques, R. V. Sabariego, C. Geuzaine, and J. Gyselinck. Inclusion of a direct and inverse energy-consistent hysteresis model in dual magnetostatic finite-element formulations. *IEEE Transactions on Magnetics*, 52(3):1–4, 2015.

[15] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. *Efficient BackProp*, pages 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[16] H. Li, S. R. Lee, M. Luo, C. R. Sullivan, Y. Chen, and M. Chen. MagNet: A machine learning framework for magnetic core loss modeling. In *2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL)*, pages 1–8. IEEE, 2020.

[17] H. J. Logarzo, G. Capuano, and J. J. Rimoli. Smart constitutive laws: Inelastic homogenization through machine learning. *Computer Methods in Applied Mechanics and Engineering*, 373:113482, 2021.

[18] E. Mušeljić, K. Roppert, L. Domenig, A. Reinbacher-Köstinger, and M. Kaltenbacher. Employing automatic differentiation and neural networks for parameter identification of an energy based hysteresis model. *International Journal of Applied Electromagnetics and Mechanics*, 73:1–13, 12 2023.

[19] N. M. Nawi, W. H. Atomi, and M. Rehman. The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology*, 11:32–39, 2013. 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013.

[20] I. Niyonzima, R. Sabariego, P. Dular, F. Henrotte, and C. Geuzaine. Computational homogenization for laminated ferromagnetic cores in magnetodynamics. *IEEE Transactions on Magnetics*, 49(5):2049–2052, 2013.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[22] F. Purnode, F. Henrotte, F. Caire, J. Da Silva, G. Louppe, and C. Geuzaine. A material law based on neural networks and homogenization for the accurate finite element simulation of laminated ferromagnetic cores in the periodic regime. *IEEE Transactions on Magnetics*, 58(9):1–4, 2022.

[23] S. Quondam Antonio, F. Riganti Fulginei, A. Laudani, A. Faba, and E. Cardelli. An effective neural network approach to reproduce magnetic hysteresis in electrical steel under arbitrary excitation waveforms. *Journal of Magnetism and Magnetic Materials*, 528:167735, 2021.

[24] J. Schröder. A numerical two-scale homogenization scheme: the fe2-method. In *Plasticity and beyond: microstructures, crystal-plasticity and phase transitions*, pages 1–64. Springer, 2014.

[25] Y. Shimizu, S. Morimoto, M. Sanada, and Y. Inoue. Automatic design system with generative adversarial network and convolutional neural network for optimization design of interior permanent magnet synchronous motor. *IEEE Transactions on Energy Conversion*, 38(1):724–734, 2023.

[26] J. Sola and J. Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3):1464–1468, 1997.

[27] S. Steentjes, F. Henrotte, C. Geuzaine, and K. Hameyer. A dynamical energy-based hysteresis model for iron loss calculation in laminated cores. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*,

27(3):433–443, 2014.

[28] P. Virtanen, R. Gommers, T. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. Walt, M. Brett, J. Wilson, K. Millman, N. Mayorov, A. Nelson, E. Jones, R. Kern, E. Larson, and Y. Vázquez-Baeza. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17:1–12, 02 2020.

[29] G. Von Pfingsten, S. Steentjes, and K. Hameyer. Operating point resolved loss calculation approach in saturated induction machines. *IEEE Transactions on Industrial Electronics*, 64(3):2538–2546, 2017.

[30] N. Vuokila, C. Cunning, J. Zhang, N. Akel, A. Khan, and D. A. Lowther. The application of neural networks to the modeling of magnetic hysteresis. *IEEE Transactions on Magnetics*, 2023.

[31] E. Weinan and E. Björn. The heterogeneous multi-scale method for homogenization problems. In *Multiscale methods in science and engineering*, pages 89–110. Springer, 2005.

[32] L. Wu, V. D. Nguyen, N. G. Kilingar, and L. Noels. A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. *Computer Methods in Applied Mechanics and Engineering*, 369:113234, 2020.

[33] H. Zhao, H. H. Eldeeb, Y. Zhang, D. Zhang, Y. Zhan, G. Xu, and O. A. Mohammed. An improved core loss model of ferromagnetic materials considering high-frequency and nonsinusoidal supply. *IEEE Transactions on Industry Applications*, 57(4):4336–4346, 2021.

□

APPENDIX

A LAMINATION MODEL

The main ingredients of the mesoscopic lamination model are given in this appendix. The lamination stack is composed of a large number of steel laminations of thickness d, galvanically insulated from each other, and stacked on top of each other. Let $\{x, y, z\}$ be a Cartesian coordinate system, with x and y in the in-plane directions of the laminations, and z in stacking direction.

The mesoscopic model is based on the geometric assumption that the variations of the magnetic field $\mathbf{h}$ in z–direction dominate the in-plane variations. This means that at each point in the stack, the lamination can be considered very large, and that only the field variation in z–direction is computed. Edge effects towards the edges of the stack are thus disregarded. Finally, all laminations in the stack see the same macroscopic $\mathbf{H}(x, y, t)$ field. The model thus solves for the unknown vector field $\mathbf{h}(z, t) = \big(h_x(z, t), h_y(z, t)\big)$ over the domain of analysis $z \in [0, d/2]$. Solving over half the lamination thickness is sufficient, due to the symmetry with respect to the median plane of the lamination.

The one-dimensional magneto-quasistatic formulation reads as follows: Find the magnetic field $\mathbf{h}(z, t)$ such that

$$\begin{cases} \partial_t \Big(b_x\left(\mathbf{h}\right), h_x' \Big)_{[0,d/2]} + \Big(\sigma^{-1} \partial_z h_x, \partial_z h_x' \Big)_{[0,d/2]} = 0 \\ \partial_t \Big(b_y\left(\mathbf{h}\right), h_y' \Big)_{[0,d/2]} + \Big(\sigma^{-1} \partial_z h_y, \partial_z h_y' \Big)_{[0,d/2]} = 0 \end{cases} \tag{A1}$$

holds for all test functions h_x', h_y'. The zero current condition through the median plane translates into the Neumann boundary conditions $\partial_z \mathbf{h}(t, 0) = 0$. The excitation of the mesoscopic model at location (x, y) is the Dirichlet boundary condition $\mathbf{h}(t, d/2) = \mathbf{H}(x, y, t)$ where $\mathbf{H}$ is the periodic macroscopic field sequence, repeated over the number of periods solved. Note that lower case letters are used for the fields in the mesoscopic model, whereas upper case letters are used for the fields in the macroscopic FE model.

The formulation (A1) accounts for eddy currents (through the terms in σ^{-1}), and for magnetic hysteresis if the material law $b_x(\mathbf{h})$, $b_y(\mathbf{h})$ implements a hystersis model, like, in our case, the energy-based hysteresis model [11, 13]. The parameters of the hysteresis model are identified from Epstein measurements performed on M235-35A steel laminations.

The domain of analysis $z \in [0, d/2]$ is discretized with 50 equidistant nodes. Time-stepping is conducted over a couple of periods with 600 time-steps per period, and the last period is retained as the periodic solution of the problem. Time derivatives are discretized with the implicit Euler scheme. Nonlinear iterations are done with the fixed-point technique. Once the mesoscopic problem has been solved, its homogenized response to the excitation $\mathbf{H}$ is the sequence $\mathbf{B}$ obtained by integration over

the half lamination thickness

$$\mathbf{B} = \frac{2}{d}\int_0^{d/2} \mathbf{b}(z,t)\,\mathrm{d}z. \tag{A2}$$

The mesoscopic model $\mathbf{B}(\mathbf{H})$ is implemented as a self-supported C language code of about 1000 lines. The linear system, whose matrix is sparse with a bandwidth equal to 3, is solved with an efficient LU sparse algorithm.

B NEWTON-RAPHSON SCHEME

The principle of the nonlinear Newton-Raphson iteration scheme is to replace $\tilde{\mathbf{H}}$ by its first order Taylor expansion in (8), to obtain

$$\left(\mu_0^{-1}\mathrm{curl}\,\mathbf{A}, \mathrm{curl}\,\mathbf{A}'\right)_{\Omega\setminus\Omega_{\mathrm{core}}} + \left(\tilde{\mathbf{H}} + \frac{\partial\tilde{\mathbf{H}}}{\partial\mathbf{B}}\Delta\mathbf{B} + \frac{\partial\tilde{\mathbf{H}}}{\partial\dot{\mathbf{B}}}\Delta\dot{\mathbf{B}}, \mathrm{curl}\,\mathbf{A}'\right)_{\Omega_{\mathrm{core}}} - \left(\mathbf{j}_s, \mathbf{A}'\right)_{\Omega_s} = 0 \tag{B3}$$

where $\Delta\mathbf{B} \equiv \mathbf{B} - \mathbf{B}^{n-1}$, with $\mathbf{B}$ the unknown field at the n^{th} nonlinear iteration, and $\mathbf{B}^{n-1}$ its value at the previous nonlinear iteration. The time derivative $\dot{\mathbf{B}}$ is approximated by the finite difference

$$\dot{\mathbf{B}} \approx \frac{\mathbf{B} - \mathbf{B}^{\mathrm{prev}}}{\Delta t}, \tag{B4}$$

with $\mathbf{B}^{\mathrm{prev}}$ the field value at the previous time-step. One has thus

$$\Delta\dot{\mathbf{B}} = \frac{\mathbf{B} - \mathbf{B}^{\mathrm{prev}}}{\Delta t} - \frac{\mathbf{B}^{n-1} - \mathbf{B}^{\mathrm{prev}}}{\Delta t} = \frac{\mathbf{B} - \mathbf{B}^{n-1}}{\Delta t}. \tag{B5}$$

By differentiating (1), one has

$$\frac{\partial\tilde{\mathbf{H}}}{\partial\mathbf{B}}(\mathbf{B}, p_k) = \left(p_0 + \left(p_1|\mathbf{B}|^2\right)^{p_2}\right)\mathbb{I} + p_1p_2\left(p_1|\mathbf{B}|^2\right)^{p_2-1}\,2\,\mathbf{B}\mathbf{B}^T \tag{B6}$$

and

$$\frac{\partial\tilde{\mathbf{H}}}{\partial\dot{\mathbf{B}}}\left(\dot{\mathbf{B}}, p_k\right) = \left(p_3 + \frac{p_4}{\sqrt{p_5^2 + |\dot{\mathbf{B}}|^2}}\right)\mathbb{I} - \frac{1}{2}\frac{p_4}{\left(p_5^2 + |\dot{\mathbf{B}}|^2\right)^{3/2}}\,2\,\dot{\mathbf{B}}\dot{\mathbf{B}}^T \tag{B7}$$

where $\mathbb{I}$ is the identity tensor and $\mathbf{V}\mathbf{V}^T$ the dyadic product, $(\mathbf{V}\mathbf{V}^T)_{ij} = V^iV^j$. Equation (B3) is then repeatedly solved until $\Delta\mathbf{B}$ reaches below a given tolerance.