Scientific machine learning applied to real-time magnetic actuator dynamic modeling

Joël MOUTERDE*, Jérôme TOMEZYK* and Joaquim DA SILVA*
*Research and Development Department, SKF Magnetic Mechatronics
2 rue des champs, 27950 Saint-Marcel, France
E-mail: joel.mouterde@skf.com

Abstract
In this article, we examine the potential of employing scientific machine learning techniques to develop an accurate dynamic real-time model of an axial electromagnetic actuator. We focus on one side of the axial AMB, specifically a fixed electromagnet with a single coil, and aim to model the magnetic flux applied to a fixed part as a function of the input voltage to the coil. Real-time capability is defined by the model's ability to operate at a frequency of at least 14Khz on a standard computer CPU. To achieve this goal, we compare two methods for modeling dynamical systems: Neural Ordinary Differential Equations (NODE), a purely data-driven approach, and Universal Differential Equations (UDE), which integrates physics-based structures into the model. In this study, the physics-based structure employed is a Cauer Ladder Network (CLN), representing the system as a linear electrical equivalent circuit. A finite element model is constructed, and the parameters of the CLN are directly extracted from the finite element matrices. The UDE discussed in this article seeks to adjust the resistances and inductances of the CLN using a neural network based on the states of the ordinary differential equation. Training and testing datasets are generated from the finite element model. Both NODE and UDE are trained using stochastic gradient descent with a quadratic loss against the training datasets, and their accuracy is evaluated using testing datasets. Our findings demonstrate that the UDE provides an accurate and stable dynamical model, delivering superior accuracy and convergence speed compared to NODE.

Keywords : Dynamical systems, Magnetic actuator, Real-time, Scientific machine learning, Neural Ordinary Differential Equations, Universal Differential Equations, Cauer Ladder Network

1. Introduction

Accurate real-time dynamic modeling of electromagnetic actuators is crucial for magnetic bearing applications, particularly in observer-based control, model predictive control, and monitoring applications. For non-laminated axial magnetic bearings, it is essential to consider the diffusive effects associated with eddy currents and the non-linearity linked to magnetic material saturation. Current state-of-the-art methods primarily utilize model order reduction techniques to derive a lower-dimensional model from a physics-based finite element (FE) model.

These methods have been previously applied to model electromagnetic actuators. For example, Proper Orthogonal Decomposition combined with Discrete Empirical Interpolation has been investigated (Jérôme Tomezyk et al., 2023) but faced stability issues and unsatisfactory computation times for real-time applications. A widely recognized technique approximates the magnetic dynamic model using a Cauer Ladder Network (CLN), representing the system as an equivalent electrical model. The resistances and inductances of the CLN can be derived directly from the FE model (A. Kameari et al., 2018). Non-linearity is addressed by interpolating the resistances and inductances based on the total current circulating within the equivalent electrical circuit (H. Eskandari et al., 2021). Although these methods are fast to compute, stable, and accurate for static points and dynamic variations around static points, they exhibit inaccuracies during large transients. An approach proposed by Chady Ghnatios et al. (2024) addresses this issue by adding a physics-informed data-driven correction to the CLN. While this method improves accuracy, it requires a high number of neural network parameters, complicating real-time implementation. Despite demonstrating the potential of physics-informed

techniques for real-time electromagnetic actuator modeling, these approaches continue to experience challenges related to computation time and/or accuracy.

Emerging techniques such as Neural Ordinary Differential Equations (NODE) (Ricky T. Q. Chen et al., 2018) and Universal Differential Equations (UDE) (Christopher Rackauckas et al., 2020) have shown promise in modeling dynamic systems, especially when focusing on global quantities of interest, such as the magnetic dynamic flux as a function of input voltage. The objective of this article is to develop models based on these techniques that are well-suited for simulating a magnetic actuator and to evaluate their ability to align with the results obtained from a computationally intensive FE method.

The article is structured as follows: first, we provide a description of the system under consideration and introduce the formulation of the finite element model. Next, we present the Linear CLN used to derive the reduced model and its nonlinear version. Subsequently, we describe the structure of the NODE and UDE analyzed in this study, along with the dataset generation process from the finite element model. Finally, we compare the accuracy and convergence speed of the two methods, leading to our conclusions.

3. System description

Axial Active Magnetic Bearings (AMBs) are generally composed of non-laminated magnetic materials, resulting in substantial eddy current effects. Consequently, these effects must be considered in transient simulations, prompting the use of the magneto dynamic formulation of Maxwell's equations. For simplicity, this analysis is limited to a single electromagnet. The vector potential A is employed, such that the magnetic flux density is expressed by $B = \mathrm{curl}A$:

$$\nabla \times (\nu(\boldsymbol{B})\nabla \times \mathbf{A}) + \sigma - iN = 0 \tag{1}$$

$$\frac{\partial \phi}{\partial t} + R_{copper} i = u \tag{2}$$

Here, N denotes the unit current density associated with the coil, i is the corresponding current, u the voltage and R_{copper} the copper resistor of the coil. The parameter ν represents magnetic reluctivity, while σ electric conductivity.

The initial simplification involves modeling the geometry as a two-dimensional axisymmetric problem, with the assumption that the rotor is stationary.

4. Finite element model

We discretize the equations (1)-(2) using the FE method. Let $\mathrm{X} \in \mathbb{R}^{N_{FE}}$ be the FE solution vector, where N^{FE} is the number of degrees of freedom. Figure 1 shows a part of the mesh and the geometry. Then, equations (1)-(2) are approximated by a finite-dimensional system:

$$M^{\nu}(X)\mathrm{X} + M^{\sigma}\frac{dX}{dt} - iN = 0 \tag{3}$$

$$N^{T}\frac{dX}{dt} + R_{copper} i = u \tag{4}$$

To discretize the temporal component, we adopt the implicit Euler method. The complete discretized problem at time step n is then formulated as follows:

$$\begin{pmatrix} M^{\nu}(X^n) + \frac{M^{\sigma}}{\delta t} & -N \\ \frac{N^T}{\delta t} & R_{copper} \end{pmatrix} \begin{pmatrix} X^n \\ i^n \end{pmatrix} = \begin{pmatrix} M^{\sigma} \\ N^T \end{pmatrix} \frac{Xn-1}{\delta t} + \begin{pmatrix} 0 \\ u \end{pmatrix} \tag{5}$$

At each time step, the resulting nonlinear system is resolved using the Newton–Raphson method, which has been implemented through an in-house solver developed in MATLAB. This problem is computationally intensive, as simulating less than one second of system behavior can require several hours, particularly when including a control loop, due to the necessity of a small time step.

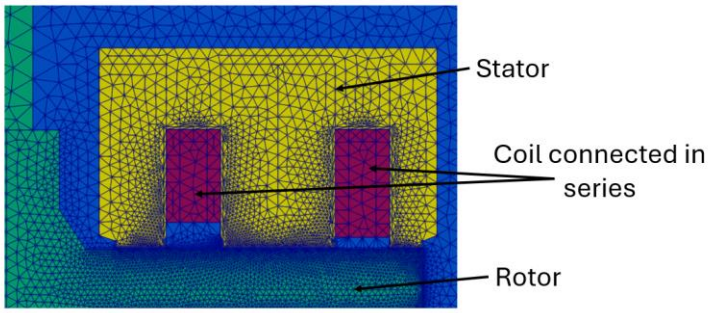

Fig. 1 *Geometry and Mesh (zoom stator part)*

5. Reduced order model definition and accuracy

The model order reduction technique utilized in this study is the CLN approach, which involves representing the FE model as an equivalent electrical circuit (cf. Figure 2). The inductances and resistances are directly derived from the FE matrices by adhering to the algorithm outlined below. A comprehensive description of the procedure is available in the works of Kameari et al. (2018) and Eskandari and Matsuo (2021).

Algorithm 1 : $K = M^{\nu}(X^{REF})$

$R_0 = 1$
$a_0 = K^{-1}N \; with \; a_0 \in \mathbb{R}^{N_{FE}}$
$L_0 = a_0^T K a_0, L_0 \in \mathbb{R}$
$e_0 = 0$
FOR $i = 2: N_{CAUER}$
  $e_i = e_{i-1} - L_{i-1}^{-1} a_{i-1}$
  $R_i = (e_i^T M^{\sigma} e_i)^{-1}$
  $a_i = K^{-1}(R_i M_i^{\sigma} + K a_{i-1})$
  $L_i = a_i^T K a_i$
ENDFOR
$R_0 = R_{copper}$

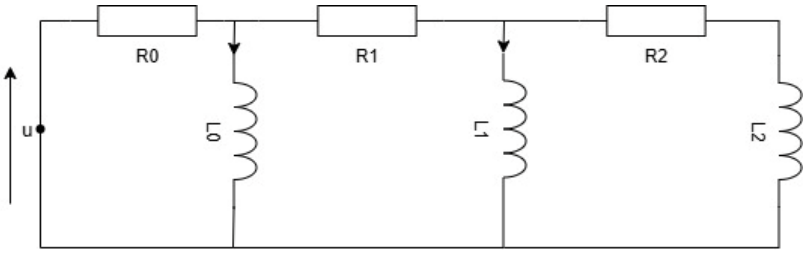

Fig. 2: Representation of the Cauer Ladder Network.

The CLN utilized here is based on a linear model. To extend the results to nonlinear scenarios, we interpolate the resistances and inductances on the total currents. Subsequently, the inductor (or resistor) corresponds to the implementation of Algorithm 1 for the solution of the magnetostatic problem, which is:

$$M^{\nu}(X^{REF})X^{REF} = iN \tag{6}$$

This approach offers the advantage of generating a reduced-order model based on the finite element problem, while relying solely on a few magnetostatic simulations. However, the resulting model can still be improved, as illustrated in Figure 7 in the results section.

6. Neural Ordinary Differential Equation model structure

NODE can be represented as:

$$h_{t_0} = \xi_{\eta}(t_0, u_0) \tag{7}$$

$$h_t = h_{t_0} + \int_{t_0}^{t} f_{\theta}(h_t, u)dt \tag{8}$$

$$\varphi_t = l_{\psi}(h_t) \tag{9}$$

where u represents the input voltage in the electromagnet, u_0, the initial voltage, h_{t_0} the initial state of the system, h_t the current state of the system, φ_t the magnetic current predicted flux, and ξ_η, f_θ, l_ψ neural networks parametrized respectively by η, θ and ψ.

In this structure, the initial state depends solely on the initial voltage. To meet this condition in practice, all simulations using this model should begin with the assumption that magnetic induction is in a static state under the specified voltage u_0.

The neural networks ξ_η and l_ψ enable the automatic selection of a latent space that best fits the system dynamics.

In this study we choose to represent f_θ as:

$$f_\theta(h,u) = NN_{\theta_1}(h) + NN_{\theta_2}(h)u \tag{10}$$

Where $\theta = (\theta_1, \theta_2)$ and NN_{θ_1}, NN_{θ_2} are neural networks respectively parametrized by θ_1 and θ_2. The linearity in the input voltage u is consistent with the physics of the system.

This study uses GELU activation functions to enhance convergence (Minhyeok Lee, 2023), except for the output layer where an identity activation function is applied.

7. Universal Differential Equation model structure

UDE is a NODE where we impose a structure on the function f_θ, which in this case derives from physics equations. Introducing this inductive bias into the neural network f_θ can enhance model training and generalization to new data not encountered during training. To construct our UDE, we suggest using the structure of the CLN, which corresponds to an electrical circuit composed of resistances and inductances. Given that the CLN is an ODE, it can be expressed in the same form as (8) (H. Eskandari et al., 2021). This implies that there exists a function $f_{CLN}(R(h), L(h), h, u)$ such that solving the CLN is equivalent to solving (8) for given *R* and *L* vectors. Then, we propose considering f_θ such that:

$$f_\theta(h,u) = f_{CLN}\left(R_{REF} + NN_{R,\theta_1}(h), L_{REF} + NN_{L,\theta_2}(h), h, u\right) \tag{11}$$

where $\theta = (\theta_1, \theta_2)$, R_{REF} and L_{REF} are fixed value vectors determined from the FE model for a given input current, NN_{R,θ_1} and NN_{L,θ_2} are neural networks used to correct respectively the base value of resistances and inductances. An example of a corrected CLN with 3 branches is represented in Figure 3. In the case of the CLN, the state h corresponds to the magnetic flux in each coil and the total flux φ_t is the flux in the first branch (H. Eskandari et al., 2021).

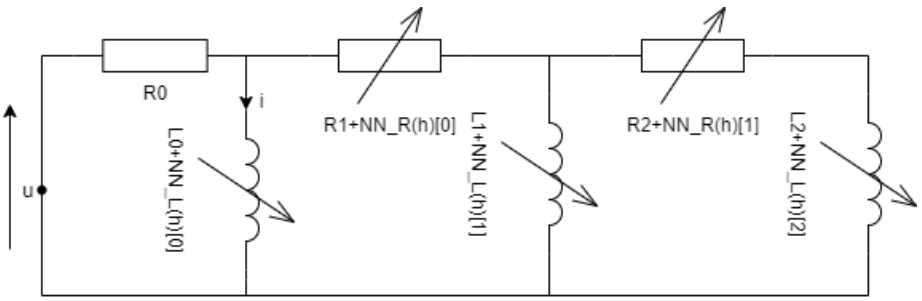

Fig. 3: Corrected Cauer Ladder Network

Given the topology of the electrical equivalent circuit, $f_{CLN}(R(h), L(h), h, u)$ can be expressed as:

$$f_{CLN}(R,L,h,u) = -\begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 1 & 1 & \ddots & & \vdots \\ 1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 1 & \cdots & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} R_1 & R_1 & \cdots & R_1 \\ 0 & R_2 & \cdots & \vdots \\ \vdots & \ddots & \ddots & R_{N-1} \\ 0 & \cdots & 0 & R_N \end{pmatrix} \begin{pmatrix} L_1^{-1} & 0 & \cdots & 0 \\ 0 & L_2^{-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & L_N^{-1} \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_N \end{pmatrix} + \begin{pmatrix} u \\ \vdots \\ u \end{pmatrix} \tag{12}$$

where R_i, L_i and h_i represents the i-th component of $R(h)$, $L(h)$ and h.

To ensure the stability of the system and remain consistent with its physics behavior, the corrected resistances and inductances should remain positive. To do so, each component i of the $R(h)$ and $L(h)$ vectors are corrected as follows:

$$R_i = R_{i\_Cauer} + R_{i\_Cauer} \cdot NN_{R,\theta_1}(h)_i \quad (13)$$

$$L_i = L_{i\_Cauer} + L_{i\_Cauer} \cdot NN_{L,\theta_2}(h)_i \quad (14)$$

where R_{i_Cauer} and L_{i_Cauer} are the i-th component of R_{REF} and L_{REF}, and $NN_{R,\theta_1}(h)_i$ and $NN_{L,\theta_2}(h)_i$ the i-th component of $NN_{R,\theta_1}(h)$ and $NN_{L,\theta_2}(h)$ outputs.

Taking an ELU activation function (Djork-Arné Clevert et al., 2016) on the NN_{R,θ_1} and NN_{L,θ_2} output layers ensures that their outputs is always > -1 which guarantees that for all i, $R_{i\_Cauer} > 0$ and $L_{i\_Cauer} > 0$. This choice ensures consistency with physics and system stability. As for the NODE, GELU activation function is used for other layers of the Neural networks.

Fixed scaling factors multiplying the networks NN_{R,θ_1} and NN_{L,θ_2} are also chosen, so that at the beginning of the training phase, the correction has no impact on the model response. This improves convergence by enabling the correction to steadily evolve to the optimal value.

8. Datasets generation

The dataset generation is a key point of the method; we decided to compute two datasets:

Low frequency dataset:

The objective of this dataset is to slowly transition from one static point ($i^0 \in \{0,1,\cdots,30\}$) to another quasi-static point, as we apply a constant voltage during a certain period of time (cf figure below). Since the system evolves slowly, a relatively large time step of 1e-3 s can be used. However, the data is then interpolated to obtain a constant time step corresponding to a 14 kHz sampling rate. This results in a total of 961 distinct simulation curves.

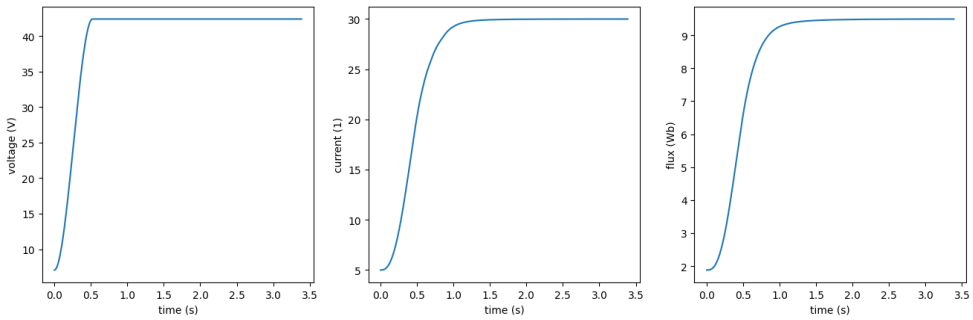

Fig. 4 Example of low frequency datatset (Voltage, Current and Flux)

High frequency dataset:

The objective of this dataset is to capture the high-frequency variations in voltage induced by the control loop. To emulate this behavior, we implement a simple proportional controller to regulate the magnetic flux. The flux setpoint is defined using a Pseudo-Random Binary Sequence (PRBS), which introduces rich frequency content and enables dynamic system excitation. We begin from a static operating point, and the flux setpoint is varied according to a PRBS with two amplitude levels. This results in a total of 62 distinct simulation curves.

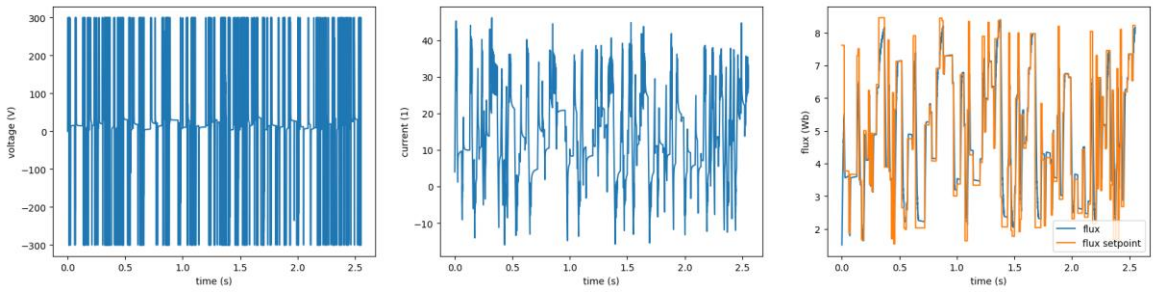

Fig. 5 Example of high frequency data (Voltage, current, flux and flux setpoint)

9. Training

The models are trained using mini-batch stochastic gradient descent on the training database with the AdamW optimizer. Automatic differentiation is used to calculate the gradients of the loss function. Each optimization iteration combines one mini-batch of high-frequency data and one mini-batch of low-frequency training data. Consequently, at each training iteration, the loss function is defined as follows:

$$\begin{cases} r_1 = \lambda_1 \sum_{i=1}^{n_{LF}} \left\| \Phi_{FEM\_LF\_i} - \Phi_{pred\_LF\_i} \right\|^2 \\ r_2 = \lambda_2 \sum_{i=1}^{n_{HF}} \left\| \Phi_{FEM\_HF\_i} - \Phi_{pred\_HF\_i} \right\|^2 \\ loss = r_1 + r_2 \end{cases} \tag{15}$$

Where n_{LF} and n_{HF} represent the low frequency and high frequency mini batches sizes, $\Phi_{FEM\_LF\_i}$ and $\Phi_{FEM\_HF\_i}$ the reference FE solutions on the entire time integration interval of the i-th low-frequency and high frequency mini-batch samples, and $\Phi_{pred\_LF\_i}$ and $\Phi_{pred\_HF\_i}$ the solutions predicted by our model on the entire time integration interval of the i-th low-frequency and high frequency mini-batch samples. λ_1 and λ_2 weighting coefficients are initiated such that the two initial loss terms are equal at the first iteration and ${\lambda_1}^2 + {\lambda_2}^2 = 1$. Then, λ_1 and λ_2 are dynamically updated at each iteration k using a residual based attention mechanism (S. J. Anagnostopoulos et al., 2024) to improve convergence:

$$\lambda_i^{k+1} = (1-\eta)\lambda_i^k + \eta \frac{r_i}{\max_i(r_i)} \tag{16}$$

Where η is chosen to adapt weighting coefficient updating speed.

The optimizer's learning rate is scheduled to decrease during training to speed up convergence.

10. Results

The state dimension is set to 15 to compare the two models' structures. All the neural networks used in the study have 3 hidden layers and 20 neurons per hidden layer. At each iteration a mini batch of 18 low frequency signals and a mini batch of 18 high frequency signals are randomly selected from the train database to compute the loss. To reduce computation time, the model's time integration is parallelized across each mini batch during training. The residual based attention mechanism parameter value is chosen at $\eta = 0.01$. The learning is chosen at 0.005 on the first 2500 iterations and reduced to 0.001 on the last 2500 iterations.

Our models' accuracy is evaluated every 10 iterations by computing the Relative Root Mean Square Error (RRMSE) over the entire time integration interval on both low and high frequency test databases. Using the test databases ensures good generalization of our model.

The expression used for the RRMSE on a mini batch is defined as follows:

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{n} \left\| \Phi_{FEM\_i} - \Phi_{pred\_i} \right\|^2}{\sum_{i=1}^{n} \left\| \Phi_{FEM\_i} \right\|^2}} \tag{17}$$

Where n represents the mini batch size, Φ_{FEM_i} the reference FE solution on the entire time integration interval of the i-th mini-batch sample and Φ_{pred_i} the solution predicted by our model on the entire time integration interval of the i-th mini-batch sample.

The RRMSE results at the end of the learning phase are given in Table 1. They show that the UDE is significantly improving accuracy in comparison with the other models.

Dataset	Cauer RRMSE	NODE RRMSE	UDE RRMSE
Low frequency test	0.00498	0.01031	0.00297
High frequency test	0.01131	0.00901	0.00130

Table 1 Model accuracy results.

The convergence of the learning phases is illustrated in Figure 6, which depicts the RRMSE averaged across both low frequency and high frequency test datasets as a function of the optimizer iteration number. This figure demonstrates that the UDE learning phase converges significantly faster than the NODE learning phase. In this example, prior to adjusting the learning rate, the NODE optimization appears to be trapped in a local minimum. The peaks observed during the initial 2500 iterations are attributable to a high learning rate, resulting in temporary instabilities during the training phase.

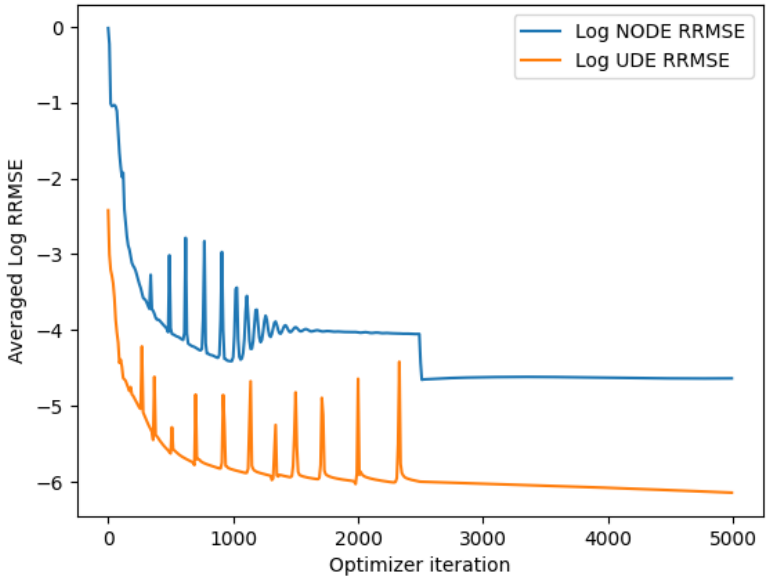

Fig. 6 Averaged Log RRMSE on test datasets

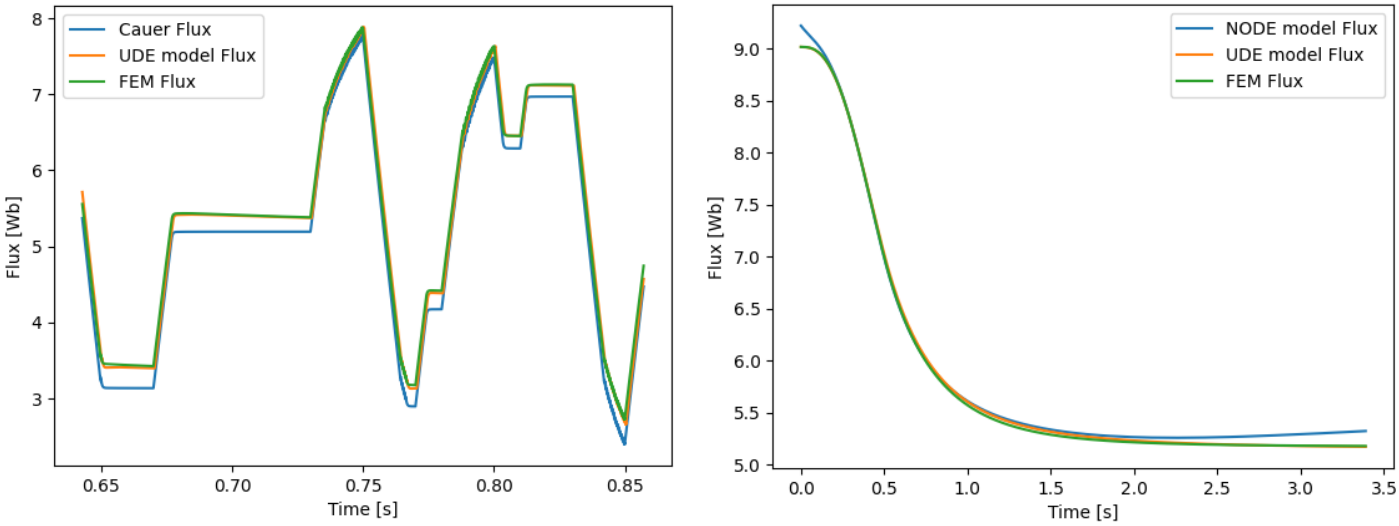

Fig. 7 Left: UDE model and CLN model results on one simulation compared to Finite Element reference. Right: UDE model and NODE model results on one simulation compared to Finite Element reference. Both results show much better accuracy of the UDE versus CLN and NODE.

The accuracy of our models is analyzed visually using the flux response versus time plots. The left part of Fig. 7 presents a zoom on a portion of a high-frequency dataset, showing that the UDE closely matches the FEM model, while the non-linear CLN model exhibits some discrepancies.

The UDE is inherently stable, while the NODE lacks stability guarantees. In the low frequency dataset shown in the right part of Fig. 7, the NODE deviates from the constant target value, whereas the UDE model closely matches the finite element solution.

The implementation was carried out using Python/JAX. The simulation time on a standard computer, equipped with an Intel(R) Core(TM) i7-13800H CPU, is approximately 0.2 seconds for the NODE model and 0.1 seconds for the UDE model for a 2.55 seconds input signal. The real-time implementation target is met with a standard CPU.

11. Conclusion

This research illustrates the advantages of applying scientific machine learning techniques to the real-time modeling of an electromagnetic actuator. The UDE model delivers high accuracy on any excitation profile, which is reportedly superior to current methods, while also enabling real-time implementation for control and monitoring applications. Future work includes predicting force complementary to the magnetic flux and extending this approach to a complete axial magnetic bearing system, considering moving parts and mutual inductances.

12. Conflicts of interest

The authors declare they have no conflict of interest regarding the content of the manuscript.

References

Jérôme Tomezyk, Sebastian Rodriguez, Yves Dupuis, Joaquim Da Silva, “Model Order Reduction of an axial magnetic bearing”, Proceedings of ISMB18, 2023

A. Kameari, H. Ebrahimi, K. Sugahara, Y. Shindo, and T. Matsuo, “Cauer Ladder Network Representation of Eddy-Current Fields for Model Order Reduction Using Finite-Element Method”, IEEE TRANSACTIONS ON MAGNETICS, VOL. 54, NO. 3, MARCH 2018

H. Eskandari and T. Matsuo, "Comparison Study of First-Order Approximations of Nonlinear Eddy-Current Field Using Cauer Ladder Network Method," in *IEEE Transactions on Magnetics*, vol. 57, no. 6, pp. 1-4, June 2021, Art no. 6300704, doi: 10.1109/TMAG.2021.3060503.

Chady Ghnatios, Sebastian Rodriguez, Jerome Tomezyk, Yves Dupuis, Joel Mouterde, Joaquim Da Silva and Francisco Chinesta, “A hybrid twin based on machine learning enhanced reduced order model for real-time simulation of magnetic bearings”, Ghnatios et al. Advanced Modeling and Simulation in Engineering Sciences (2024) 11:3 https://doi.org/10.1186/s40323-024-00258-2

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud, “Neural Ordinary Differential Equations”, https://arxiv.org/abs/1806.07366, 2018

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, Alan Edelman, “Universal Differential Equations for Scientific Machine Learning”, https://arxiv.org/abs/2001.04385, 2020

Minhyeok Lee, “GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance”, 2023, https://arxiv.org/abs/2305.12073

Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)”, ICLR 2016, https://arxiv.org/abs/1511.07289v5

S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopulos, and G. E. Karniadakis, “Residual-based attention in physics-informed neural networks,” Comput. Methods Appl. Mech. Eng., vol. 421, p. 116805, Mar. 2024. DOI: 10.1016/j.cma.2024.116805.