Fast Frequency-Domain Analysis for Parametric Electromagnetic Models Using Deep Learning

Elia Mattucci
Rete Ferroviaria Italiana S.p.A.
Direzione Tecnica - Standard Tecnologie
Rome, Italy
e.mattucci@rfi.it

Lihong Feng, Peter Benner
Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany
{feng,benner}@mpi-magdeburg.mpg.de

Daniele Romano, Giulio Antonini
Dipartimento di Ingegneria Industriale e dell'Informazione e di Economia
University of L'Aquila
L'Aquila, Italy
{daniele.romano,giulio.antonini}@univaq.it

***Abstract*—The design of microwave devices relies on the modeling and simulation of mathematical models of such devices. Usually, the mathematical models are very large, causing long simulation times and making the design process very slow. In this work, we propose a deep learning method for fast frequency-domain simulation of parametric models of microwave devices. We focus on the fast evaluation of the parameter-dependent transfer function via approximating the transfer function with a neural network. After being properly trained using limited data of the transfer function, the trained neural network can evaluate the transfer function at any parameter sample with acceptable accuracy. The proposed deep learning method is tested on a PEEC model of a microwave device and shows its efficiency in fast predicting the transfer function.**

***Index Terms*—Deep learning, Partial Element Equivalent Circuit (PEEC) method, fast transfer function prediction.**

I. Introduction

Frequency-domain analyses are often implemented in computational electromagnetics for design purposes. This is usually done by simulating large mathematical models in the frequency domain, in particular, by computing the transfer functions at many frequencies in a large frequency interval. Moreover, to achieve the goal of optimal design, the response of the system under the influence of design parameters needs to be analyzed. As a result, the parameter-dependent transfer function is repeatedly computed at many different values of the parameters and different frequencies. The computation is expensive when the spatially discretized finite element models or the PEEC models are of large dimensions.

Parametric model order reduction (PMOR) methods have been proposed to reduce computational costs while keeping the output response or transfer function sufficiently accurate in the whole parameter and frequency/time domain [1], [2]. In this work, we consider systems with geometrical parameters and seek methods for fast transfer function evaluation while both the frequency and the parameters change in certain ranges. We don't require that the system matrices are explicitly available. Instead, only data from the transfer function are needed. In this sense, we are considering non-intrusive PMOR. MOR methods based on Loewner framework use only the data of the transfer function to build the reduced-order model and the reduced transfer function [3], [4]. Non-intrusive PMOR methods based on pole-matching are proposed in [5], etc. The above-mentioned non-intrusive PMOR methods use a rational function to approximate the transfer function. In some cases, the transfer functions are non-rational, for example, the transfer function of time-delay systems with many delays. Neural transfer function based PMOR methods are proposed in [6], [7], which nevertheless require that the original system matrices are available to compute the reduced frequency-domain responses, therefore may not applicable to measured data.

Recently, machine learning is used to predict the transfer function for non-parametric time-delayed PEEC models [8]. The method proposed there has no limitation on the transfer function, i.e., non-rational transfer functions of time-delay systems with many delays can also be accurately predicted by the proposed neural networks. This may not be achieved by the rational function based methods [6], [7], [9]. Furthermore, the method in [8] does not require the original system matrices to be available, which can be easily applied to measured data.

In this work, we consider applying deep learning methods to enable fast frequency-domain analysis for parametric PEEC models. A deep feed-forward neural network (DFNN) was used in [8], where a DFNN is established for each entry of the transfer function matrix. We modify the structure of the DFNN so that a single DFNN can predict all the entries of the transfer function, without inducing much extra offline training time. In this way, we save a lot of training time for systems with multiple inputs and multiple outputs. Furthermore, the proposed single DFNN is able to predict transfer functions under physical and geometrical parameter variations. In contrast to the existing neural transfer function based PMOR methods [6], [7], where the NNs are trained to learn the poles/residues or the coefficients of polynomials in the assumed rational representation of the transfer functions, we use DFNN to directly learn the values of the transfer functions. Consequently, the proposed DFNN naturally avoids the pole-residue mismatching issues in [6], [7], [9]. Another advantage of the proposed DFNN is that only one set of training data with one time of training is sufficient to achieve accurate transfer function prediction, whereas two sets of training data from vector fitting/reduced-order models and the

original electromagnetic simulation, respectively, are needed to train a neural network for two times in [6], [7], [9].

The next section presents parametric n-port electromagnetic systems in the frequency domain. Then we introduce the structure of DFNN used for transfer function prediction. Simulation results are detailed in Section IV. Finally, conclusions are given.

II. Parametric n-port electromagnetic systems

It is known that frequency-domain analysis of electromagnetic systems is usually done via computing the Z, Y, or S parameters at discrete frequency points in the frequency range of interest. During the optimization and design phases, finding the link between transfer (function) matrices and geometrical parameters could be very useful in reducing design times. For this reason, mathematical models of the parameterized n-port electromagnetic systems are repeatedly simulated at many samples of the parameters and frequencies. The model in the frequency domain is in the form of a transfer function:

$$\boldsymbol{H}(s,\boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu})(\mathbf{G}(s,\mu))^{-1}\mathbf{B}(\boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu}), \qquad (1)$$

where $\mathbf{G}(s,\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}, \mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{N\times n}, \mathbf{C}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times N}$ and $\mathbf{D}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}$. $\mu \in \mathbb{R}^p$ is the vector of parameters. Since the system has n ports, $\boldsymbol{H}(s,\boldsymbol{\mu})$ is a matrix with n rows and n columns. N is often very large, making the computation of $\boldsymbol{H}(s,\boldsymbol{\mu})$ very slow. Depending on the problem considered, $\boldsymbol{H}(s,\boldsymbol{\mu})$ can be rational or non-rational. The goal of this work is to find a generic model able to fast reproduce the frequency behavior of the system when the parameters change. Artificial neural networks can help with this need: after an initial effort to generate data for training, the trained neural network (NN) can be expressed as an operator mapping the parameter and frequency to an output: $\boldsymbol{g_\theta} : (\boldsymbol{\mu}, f) \mapsto \tilde{\boldsymbol{H}}(\boldsymbol{\mu}, f)$:

$$\tilde{\boldsymbol{H}}(\boldsymbol{\mu}, f) = \boldsymbol{g_\theta}(\boldsymbol{\mu}, f). \qquad (2)$$

Here, $\tilde{\boldsymbol{H}}$ is the output vector of the NN with dimension dependent on the number of ports n and the type of the NN. $\boldsymbol{\theta}$ is the vector of hyper-parameters of the neural network, e.g., the number of hidden layers and units.

III. Deep Learning for parameterized frequency-domain simulation

In this work, we propose to apply a DFNN for parametric transfer function prediction.

A. DFNN structure

DFNN is also called multilayer perceptrons (MLP). The input data are fed into the NN, then are weighted and summarized. A weighted sum of the input vector plus a bias is then taken as the argument of an activation function $\boldsymbol{g}$, which completes the first hidden layer. The output of the first hidden layer is then fed into the next hidden layer, and the same summation and activation then continue. The final layer gives the output of the NN. Usually, there are no activation functions on the output layer. The network can be described as:

$$\tilde{\boldsymbol{H}} = W_\ell^T \boldsymbol{g}(\cdots \boldsymbol{g}(W_1^T \boldsymbol{g}(W_0^T \boldsymbol{x} + \boldsymbol{b}_0) + \boldsymbol{b}_1)) + \boldsymbol{b}_\ell, \qquad (3)$$

where $\boldsymbol{x}$ is the vector of inputs. $W_i \in \mathbb{R}^{n_i \times n_{i+1}}, \boldsymbol{b_i} \in \mathbb{R}^{n_i}, i \geq 0$ are the weight matrices and bias vectors. $\boldsymbol{g} : \mathbb{R}^{n_i} \mapsto \mathbb{R}^{n_{i+1}}$ is the vector of activation functions. Typically, $\boldsymbol{g} = (g, \ldots, g)$ and $\boldsymbol{g}(\tilde{\boldsymbol{x}}) = (g(\tilde{x}_1), \ldots, g(\tilde{x}_{n_i}))$. $g : \mathbb{R} \mapsto \mathbb{R}$ is an activation function, and $\tilde{x}_i$ is the i-th entry of $\tilde{\boldsymbol{x}}$. In other words, the activation function is applied to the weighted input vector element-wise. The weights and biases on all the layers of NN are to be optimized during the training of the NN.

B. Data preparation

Correct formulation of the data plays a key role for the success of deep learning. We explain how to prepare the input-output data for the DFNN. The PEEC models have the property of reciprocity leading to symmetric transfer matrices. Therefore, for an n-port model, we only need to consider $n \cdot (n+1)/2$ outputs, so that only $n \cdot (n+1)/2$ entries of the transfer matrix need to be predicted by the NN. Since the activation functions of an NN are all real-valued functions, we let the NN predict the imaginary part and real part of the transfer matrix separately. However, the real and imaginary parts can be predicted simultaneously.

As for DFNN, the vector of input fed into DFNN is $(\mu_1, \ldots, \mu_p, f)$, so that the dimension of the input fed into the DFNN is $p+1$. The output is a vector of $n \cdot (n+1)$ entries including the real and imaginary part of the transfer matrix corresponding to the $n\cdot(n+1)/2$ ports. As a result, the dimension of the DFNN output layer is set as $n\cdot(n+1)$. When training the DFNN, the input data can be seen as a matrix of $n_\mu \cdot n_f$ rows and $p+1$ columns, since we train DFNN using $n_\mu \cdot n_f$ samples of the input vector. Here, n_μ is the number of parameter samples and n_f is the number of frequency samples. The output data corresponding to the training input data is also a matrix of $n_\mu \cdot n_f$ rows but $n \cdot (n+1)$ columns. When using the trained DFNN for prediction, any combination of the samples of μ and the sample of the frequency, i.e. $(\mu_1^{i_1}, \ldots, \mu_p^{i_p}, f_j)$ can be fed into DFNN, from which we can get the predicted imaginary and real parts of the transfer matrix entries at $(\mu_1(i_1), \ldots, \mu_p(i_p), f_j)$. To train the DFNN, we use a mean squared error (MSE)-based loss function.

IV. Numerical Results

The proposed approach has been tested on the parameterized model of a Wireless Power Transfer System shown in Fig. 1. We consider five geometrical quantities as parameters: the width b of a single conductor, the thickness h of a single conductor, the distance D between coils and the lengths L_1, L_2. We fixed $n = 2$ electrical ports, explicitly visible in Fig.1 too.

To generate the data for training the neural network, we compute the S-parameters at 20 frequency samples in $[5, 30]$MHz. The parameters have central points: $b = 1\cdot10^{-3}$m, $h = 1 \cdot 10^{-3}$m, $L_1 = 88.6 \cdot 10^{-3}$m, $88.6 \cdot 10^{-3}$m and $D = 0.01$m, and vary with $\pm10\%$. We take 3 samples for each parameter, then take the tensor product of them, resulting in total 3^5 samples of [b, h, L_1, L_2, D]. All parameter samples are further combined with the 20 frequency samples

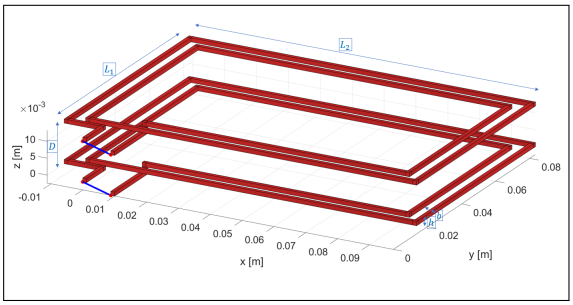


Fig. 1. WPT System.

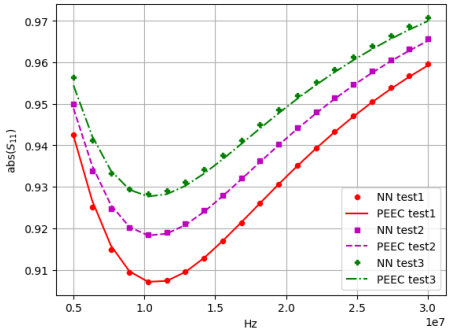
to obtain $3^5 \cdot 20 = 4860$ samples of the six input variables (f, b, h, L_1, L_2, D) for the DEFF. We take 4660 samples as training data and 100 samples as the validation data. The remaining 100 samples are the testing data. For DFNN, the dimension of the output is $n \cdot (n+1) = 6$.

The DFNN has 6 hidden layers with 10 neurons on each layer. The sinusoidal function in TensorFlow: tf.math.sin is taken as the activation function function, which provides better results than other activation functions. After 2422.79s with 2000 epochs, the network was trained and able to predict in 0.00501s the S-parameters at all the testing parameters and frequencies. The training time could be largely reduced when we use a more efficient network structure, e.g., autoencoder, convolutional network. This will be the future work.

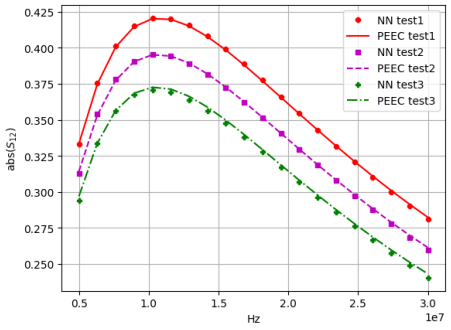
Predictions for test values are shown in Fig. 2, where testing geometrical samples are $b = 9 \cdot 10^{-4}$m, $h = 9 \cdot 10^{-4}$m, $L_1 = 79.7 \cdot 10^{-3}$m, $L_2 = 79.7 \cdot 10^{-3}$m, $D = 0.009$m for test 1, $b = 9 \cdot 10^{-4}$m, $h = 9 \cdot 10^{-4}$m, $L_1 = 79.7 \cdot 10^{-3}$m, $L_2 = 79.7 \cdot 10^{-3}$m, $D = 0.01$m for test 2, $b = 9 \cdot 10^{-4}$m, $h = 9 \cdot 10^{-4}$m, $L_1 = 79.7 \cdot 10^{-3}$m, $L_2 = 79.7 \cdot 10^{-3}$m, $D = 0.011$m for test 3. "NN test$i$", $i = 1, 2, 3$, are the predicted values of the S-parameters and "PEEC test$i$", $i = 1, 2, 3$, are those obtained from direct simulation. These show that DFNN is accurate in predicting the magnitudes of both S-parameters at 3-testing parameters. Similar accuracy is also achieved by DFNN at other testing parameters and for the phases of the S-parameters.

V. Conclusion

In this work, we proposed a fast frequency domain simulation method based on deep forward neural network. After being trained using limited data, the proposed DFNN can fast and accurately predict the transfer function at any testing parameter. A main novelty of the proposed method is that it does not use any rational formulation to approximate the transfer function, and therefore is also accurate for non-rational transfer functions. The proposed approach may not guarantee accurate transient simulation. Accurate transient simulation could be studied in the future by customizing the loss function to include the internal relation between the real and imaginary parts of the transfer function.



(a) Magnitude of S_{11} from port 1 to port 1.



(b) Magnitude of S_{12} from port 2 to port 1.

Fig. 2. Example 1: DFNN prediction for S_{11} and S_{12}.

References

[1] A. Zanco and S. Grivet-Talocia, "Toward fully automated high-dimensional parameterized macromodeling," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 9, pp. 1402–1416, 2021.

[2] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM Rev.*, vol. 57, no. 4, pp. 483–531, 2015.

[3] Y. Q. Xiao, S. Grivet-Talocia, P. Manfredi, and R. Khazaka, "A novel framework for parametric Loewner matrix interpolation," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 12, pp. 2404–2417, 2019.

[4] A. C. Antoulas, C. A. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*, ser. Computational Science & Engineering. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020.

[5] Y. Yue, L. Feng, and P. Benner, "An adaptive method for interpolating reduced-order models based on matching and continuation of poles," in *2019 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO 2019)*, 2019.

[6] J. Zhang, F. Feng, W. Zhang, J. Jin, and Q.-J. Zhang, "A novel training approach for parametric modeling of microwave passive components using Padé via Lanczos and EM sensitivities," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 6, pp. 2215–2233, 2020.

[7] J. Zhang, J. Chen, W. L. Q. Guo, F. Feng, and Q.-J. Zhang, "Parameterized modeling incorporating MOR-based rational transfer functions with neural networks for microwave components," *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 5, pp. 379–382, 2022.

[8] L. Feng, P. Benner, D. Romano, and G. Antonini, "Matrix-free transfer function prediction using model reduction and machine learning," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 12, pp. 5392–5404, 2022.

[9] Y. Zhuo, F. Feng, J. Zhang, and Q.-J. Zhang, "Parametric modeling incorporating joint polynomial-transfer function with neural networks for microwave filters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 11, pp. 4652–4665, 2022.