Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082254

ARTICLE

Differential Evolution-Based Extraction of Impedance Parameters for Wide-Band Equivalent Circuits

Piotr Musznicki[1], **Marek Turzyński**[1], **Lyu Guanghua**[2], **Ghulam E Mustafa Abro**[3,*], **Viola Gierszewska**[1], **Arsalan Muhammad Soomar**[1] **and Syed Hadi Hussain Shah**[2]

[1]Department of Power Electronics and Electrical Machines, Gdańsk University of Technology, Gdańsk, Poland
[2]PowerChina Huadong Engineering Corporation Limited, Hangzhou, China
[3]Research Unit for Robophilosophy and Integrative Social Robotics (RISR), Aarhus University, Aarhus C, Denmark

*Corresponding Author: Ghulam E Mustafa Abro. Email: mustafa.abro@ieee.org

Received: 12 March 2026; Accepted: 03 July 2026; Published: 13 August 2026

ABSTRACT: This paper presents an accurate and efficient methodology for parameter extraction in complex impedance models using Differential Evolution (DE), an evolutionary optimization technique. The proposed approach targets equivalent RLC circuit topologies and aims to match measured impedance characteristics across a wide frequency spectrum. By formulating the extraction process as a global optimization problem, DE enables precise identification of component values, even for high-order models with multiple resonances. The method is implemented in Python using open-source libraries, facilitating reproducibility and integration into broader modeling workflows. Validation is performed on both analytically derived resonant circuits and physically representative systems, confirming the method's reliability in identifying parameter sets that accurately reproduce experimental impedance behavior. The results highlight the method's robustness, scalability, and suitability for automated impedance modeling in energy-related applications.

KEYWORDS: Circuit topology; equivalent circuits; evolutionary algorithms; impedance parameter extraction; wide-band modeling

1 Introduction

Nowadays, modeling and simulation have become an essential part of designing and optimizing power electronics converters that operate with high switching frequency. These tools are also advantageous when it comes to selecting filters or mitigating electromagnetic interference (EMI). However, there is still a challenge in specifying parameter values for accurate replication of real device behavior. Especially wide-band models, apart from the basic operational parameters, contain parasitic components that are difficult to identify. This paper explores a new approach of extracting passive components by using an impedance characteristic in the function of frequency $Z = f(f)$ based on the DE algorithm.

The analytical and physical methods of determining model values require geometrical properties or an accurate interpretation of device behavior and physical phenomena. Moreover, often, selected components contribute to several phenomena, which makes it difficult to determine their exact value. In the method presented, only the model's topology and impedance response (measured via an impedance analyzer) are required. The main focus of this paper is to demonstrate how the DE algorithm can be used to extract model parameters through a few representative examples in wideband circuit modeling with a switching mode power supply.

The paper includes a brief literature review on the methods of extracting impedance parameters 1.1, followed by an explanation of the fundamentals and examples of applying the differential evolution algorithm. Section 3 outlines the procedure for building RLC models based on resonance analysis and describes how the method can be applied in the Python environment. Lastly, Sections 4 and 5 present examples of using the parameter extraction method with data from simulations and impedance bridge measurements. A brief conclusion is provided at the end.

In the case of simple circuit models that consist of only a few components such as resistors, inductors, and capacitors, it is possible to calculate parameters directly using the experimental frequency characteristic $Z(f)$ of the modeled object in the real state. This methodology is described in publications such as [1,2]. In the considered frequency range, modeled impedance may unambiguously have a resistive, inductive, or capacitive character, indicating a model component responsible for $Z(f)$ the characteristic. Hence, by relating the impedance value to the corresponding frequency value, the considered model's component parameter may be easily calculated [3,4]. Moreover, parameters of the model's components may also be obtained by analyzing resonant frequencies distinguished in the experimental $Z(f)$ characteristic [4–6]. The usefulness of this model parameter extraction method is highly limited by the level of model complexity, despite its high efficiency. In the case of more complicated circuit models, other approaches are proposed. For example, in [7], there is a presented modeling of a common-mode EMI equivalent circuit model for a drive inverter. The final model comprises individual models of the motor, heat sink, cables, etc. The topology of these models is highly simplified, which allows parameterizing based on experimental frequency characteristics measured irrespective of each component of the final model. This approach seems to be an attractive alternative. However, it should be noted that due to mutual interactions between each part of the total model, the final simulated impedance frequency response may be significantly different from the experimental one. Another solution is presented in [7,8], where the model's component parameters are extracted based on a series of experimental $Z(f)$ characteristics measured for different modelled object configurations. It means that during the measurement, some object's terminals are short-circuited, and other ones are opened at the same time (similar to the common-mode (CM) and differential mode (DM) impedance measurement procedure of induction motors [6]). Such an approach allows for the elimination of the impact of some components on the Z (f) characteristic during the measurement performed for the selected configuration, which enables the determination of the rest of the components' parameters. Application of the method brings satisfying results. However, it should be noted that the usefulness is limited by the complexity of the model topology and the number of parameters that must be parameterized. For certain complex models, such as those used for motors measured in CM and DM modes, certain parameters can have a significant impact on the $Z(f)$ characteristic, making it difficult to determine the exact values of these parameters, regardless of the object configuration during the measurement process. An optimization approach is proposed to address the issue of parameterizing model parameters. In [9,10], optimization methods based on genetic algorithms, Wiener filtering, or the sum of squared errors criterion have been used to extract the model's parameters. The resulting values confirm the effectiveness of this approach. However, optimization methods often require a prior symbolic expression of the absolute value of model impedance as a function of frequency, which forces operation in a domain of complex numbers [11]. Final mathematical expressions describing the $Z(f)$ relation may have a complicated form, which increases the likelihood of errors and makes the derivation process time-consuming, especially for complex models. Knowledge of accurate wide-band model values allows for the selection of energy conversion system components at an early design stage, as well as for the mutual adaptation of converters, machines, and filters to reduce the level of electromagnetic interferences.

Despite the popularity and effectiveness of DE in various optimization tasks, there is a lack of published articles specifically exploring its application for parameter extraction of passive elements based on impedance response. Many articles contain a method of extracting the value of passive components using frequency-impedance response. The majority of these articles focus on simple models in which the first resonance peaks are identified. In this paper, the method of the impedance extraction of more complex models contains more resonances whose frequencies depend on numerous components, and each component affects several resonances, making their extraction much more difficult.

This paper is organised as follows, with the first section serving as an introduction to the main body of the study. The principles of differential evolution algorithms are discussed in Section 2, which serves as a literature survey. The approach and the topology of the model are described in Section 3. In the fourth section, the simulation is described, and in the fifth section, the verification of the impedance model formulation and implementation in Python is discussed. This section gives a comparison with other optimisation approaches that are currently in use. The paper comes to a close with the final part.

2 Differential Evolution Algorithm

DE is a popular opinion-based metaheuristic algorithm that optimizes problems over continuous domains; it is a competitive evolutionary algorithm (EA) that excels at solving complex numerical optimization problems. DE is an evolutionary approach for generating real-valued multi-modal functions that are powerful and easy [12]; DE algorithms have been widely used to solve optimization problems [13]. An evolutionary process optimizes an optimal solution iteratively [14]. Evolutionary operators operate on an evolutionary algorithm population, attempting to generate keys with increasing fitness. Initially proposed by Price and Storn, the DE algorithm is a robust, population-based optimization method that excels in handling nonlinear, non-differentiable, and multi-modal objective functions. Due to its parallelizability, the field attracted the attention of serious academics. DE is a potent tool for stochastic optimization, and modern computational techniques have only bolstered this. DE is an EA that optimizes mathematical problems across multiple dimensions. This algorithm refines its candidate solution iteratively until it achieves the highest possible value of the user-defined objective function to find the best possible solution to the problem [15].

DE was initially designed to meet some requirements that have made it particularly useful:

1. Ability to handle non-differentiable, nonlinear, and multimodal cost functions.
2. The ability to parallelize to optimize computationally intensive cost functions.
3. Ease of use is designed to be user-friendly with only a limited number of control variables required to direct the minimization process. These variables should also be robust and straightforward to choose.

It creates new candidate solutions by combining the three major operations of mutation, crossover, and primary selection [16]. It has a simple implementation but a high problem-solving quality, making it one of the most popular population-based algorithms, with several reported techniques applied. The basic structure of a DE algorithm is shown below in Fig. 1.

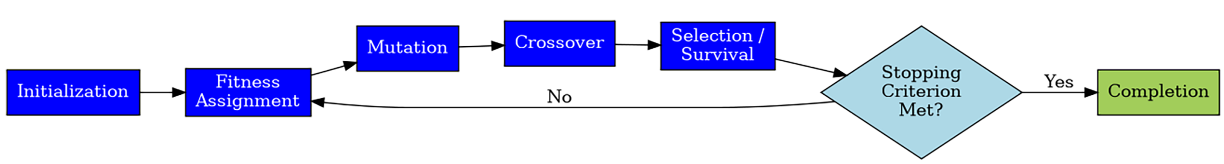


Figure 1: The basic structure of a DE algorithm.

A population of potential solutions, represented as vectors of real numbers, is initialized by the DE method. The complexity of the issue and the available processing power are typically considered when determining the population size. The algorithm then goes through several generations, each comprised of the following actions:

- **Mutation:** In this stage, a new candidate solution is created by combining a third randomly selected solution with the difference between two selected solutions. The magnitude of the mutation is determined by multiplying the difference by a scaling factor. After going through the process, each solution in the population generates a fresh set of mutant solutions, resulting in a new population [17].
- **Crossover:** In this stage, each mutant solution is mixed with the associated parent solution to create a trial solution. How the two answers are joined is determined by the crossover operator. The binomial crossover is a typical operator that switches the corresponding values between the mutant and parent solutions at random places in the solution vector [18].
- **Selection:** The trial solution is contrasted with its corresponding parent solution in this stage. If the trial solution is better in the next generation, it replaces the parent solution; otherwise, the parent solution is kept [19].

Differential Evolution uses the mutation operator in each generation (iteration) G to construct the donor vector v_i for each x_i in the present population. For each target vector $x_i = (x_i, 1, \ldots, x_i, D)$ at generation G, the associated mutant vector $v_i = (v_i, 1, \ldots, v_i, D)$ can be produced using a specific mutation scheme. The six most widely used mutation schemes [20] from Eqs. (1)–(6) in differential evolution are described below:

DE/rand/1

$$v_i = x_{r1} + F\left(x_{r2} - x_{r3}\right) \tag{1}$$

DE/rand/2

$$v_i = x_{r1} + F\left(x_{r2} - x_{r3}\right) + F\left(x_{r4} - x_{r5}\right) \tag{2}$$

DE/best/1

$$v_i = x_{best} + F\left(x_{r1} - x_{r2}\right) \tag{3}$$

DE/best/2

$$v_i = x_{best} + F\left(x_{r1} - x_{r2}\right) + F\left(x_{r3} - x_{r4}\right) \tag{4}$$

DE/current-to-best/1

$$v_i = x_i + F\left(x_{best} - x_1\right) + F\left(x_{r1} - x_{r2}\right) \tag{5}$$

DE/current-to-best/2

$$v_i = x_i + F\left(x_{best} - x_i\right) + F\left(x_{r1} - x_{r2}\right) + F\left(x_{r3} - x_{r4}\right) \tag{6}$$

From the above Eqs. (1)–(6), the indices r_1, r_2, r_3, r_4 and r_5 are random integers that are mutually exclusive and within the range $[1, NP]$, and are also different than index $i (r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i)$. For each mutant vector, these indices are created only once. The best individual vector x_{best} is defined as the member of the population at generation G with the highest objective value. The scaling factor F is a positive control parameter for scaling the difference vector. Here, x_i is the target vector at generation, G x_{best} is the best individual of the current generation, $r_1, \ldots, r_n$ are distinct randomly selected indices; and F is the scaling factor. In the DE/current-to-best/2 strategy, the donor vector is generated using the best individual of the current generation in addition to difference vectors. The function $rand(.)$ denotes a uniformly distributed random number in $[0, 1]$.

The DE algorithm generates new candidate solutions through mutation and crossover and selects the best answers from each generation until a stopping condition is met. The stopping criteria can either be a maximum number of iterations, a maximum number of function evaluations, or a fitness value threshold. DE is simple to implement and has a few impactful parameters, such as population size, scaling factor, and crossover rate. DE has been modified and extended in numerous ways, including multi-objective DE, self-adaptive DE, and hybrid DE with other optimization approaches [21].

Related Work

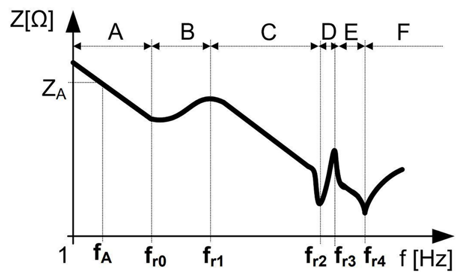
The DE algorithm has a few parameters and specific characteristics, such as optimal value accuracy and convergence rate, that significantly impact algorithm performance. The parameters still need to be determined reliably. Researchers have made some attempts to address this issue; however, choosing the DE control settings is more complex than expected [22]. The values of its parameters influence the performance of the DE algorithm. It is important to vary the parameter settings while testing different functions. Currently, three primary methods exist for specifying parameters: (1) Adaptive parameter setting: heuristic criteria are used to change parameter values based on the current state. (2) Determined parameter setting technique: the method is primarily determined by experience, such as maintaining a constant value throughout the evolutionary process. (3) Self-adaptive parameter setting: Using evolution of evolution to perform self-adaptive parameter settings. The authors proposed a fuzzy adaptive parameter setting approach that can dynamically adjust the parameters. The experiment shows that F and CR result in significantly faster convergence than the traditional DE technique. Self-adapting control parameters in DE are proposed, and the results show that the revised algorithm outperforms, or is on par with, the conventional DE method [22].

DE has been successfully applied to various domains such as antenna design, electromagnetic compatibility, and communication systems in the context of wideband modeling. In antenna design, DE optimizes the geometry of Ultra-Wideband (UWB) antennas, resulting in improved impedance matching and radiation pattern performance without providing initial assumptions. In electromagnetic compatibility, DE provides wide-band equivalent circuit models of passive components, resulting in precise frequency-domain predictions. In communication systems, DE has been used to optimize wide-band filters and amplifiers, resulting in exceptional bandwidth and gain. The Differential Evolution algorithm has been proven to be effective in wideband modeling research, producing comprehensive and valid results in various domains, as references are in [22,23]. It is an algorithm that enhances global search capabilities by utilizing historical search information. The proposed algorithm outperforms benchmark functions and shows promise in optimal chemical process control.

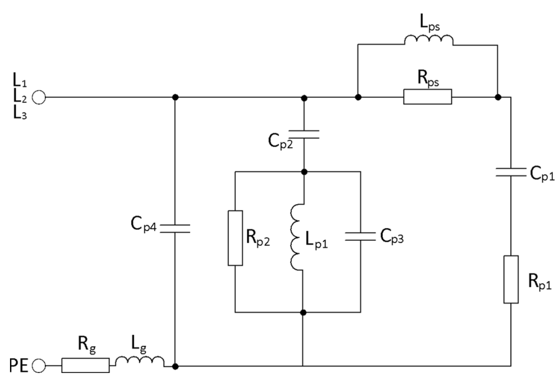
DE has been applied to optimize complex engineering systems, including circuit design and parameter estimation, showing effectiveness in frequency-constrained electrical circuits [24]. A two-stage DE with a novel mutation strategy improves parameter estimation accuracy, suitable for complex system modeling such as electronic circuits [21]. An improved DE with novel mutation and random frameworks has been validated for optimization tasks, extendable to EMI filtering and circuit parameter extraction [25]. Even if an EMI filter's insertion loss meets standards, changes in source/load impedance can reduce effectiveness. An improved differential algorithm has been proposed to address this limitation. A DE-based optimization tool enables automated EMI filter design, considering component, cost, space, and weight constraints, with measurements confirming sufficient attenuation within specified limits.

3 Methodology and Model Topology

The typical experimental impedance frequency response $Z(f)$ covers several frequency ranges in which the impedance shows a capacitive, inductive, or resistive character (Fig. 2a). At the boundaries of regions where impedance changes its character from inductive to capacitive (or *vice versa*), the series or parallel resonance process is excited. To reflect $Z(f)$ frequency characteristics in a wide range of frequencies (from single Hz up to tens of MHz), wide-band circuit models have to be applied (Fig. 2b). Complex circuits composed of capacitors, inductors, and resistors are connected in series and parallel, resulting in a complex frequency range with multiple resonances. In dependence on the complexity level, the analytical description of these models requires applying the domain of complex numbers, which requires using mathematical operations of addition, multiplication, calculation of absolute values, and real and imaginary parts of complex numbers, describing the resultant model impedance as a function of frequency. As a result, the final analytical equations may have a complex structure, making the extraction process time-consuming and increasing the risk of errors as the model becomes more intricate.



(**a**)



(**b**)

Figure 2: (**a**) An exemplary frequency characteristic of inductor motor common-mode impedance and (**b**) a wideband circuit model of inductor motor common-mode impedance.

To simplify the model topology construction process, it can be assumed that in the selected frequency range, impedance response is determined by selected model components while omitting other components' influence. For example, considering the model and characteristics presented in Fig. 2, it can be assumed that in a range indicated by the letter F, impedance $Z(f)$ has an inductive character, so for the model presented in Fig. 2b, this part of the characteristic $Z(f)$ may be modeled by a series connection of R_g and L_g. Next, for frequency f_{r4}, a series resonance between L_g and resultant capacity is observed; hence, on the boundary between regions E and F, characteristics $Z(f)$ change character from inductive to capacitive. Thus, in region E, impedance may be modeled by a capacitance, which is formed by a series-parallel connection of capacitors C_{p2}, C_{p3} and C_{p4}. By applying this approach, as the frequency f is decreased, the model can be expanded by adding more capacitive or inductive elements, which are responsible for the modeling of $Z(f)$ characteristics in subsequent regions A–D. At resonant frequencies, matching simulated $Z(f)$ characteristics to the experimental one requires including additional resistors R_{p1}, R_{p2} and R_{ps} changing the model topology.

Parameters of capacitors and inductors forming the considered model may be calculated from the known values of resonant frequencies. Also, if the width of a particular frequency region is sufficient to unambiguously assess the impedance character in that region (resistive, capacitive, or inductive without intermediate combinations), the value of resultant resistance, inductance, or characteristic in this frequency range may be calculated directly from the measured $Z(f)$ characteristic as it is presented for impedance Z_A measured for frequency f_A (Fig. 2a). However, extracting resistor values from the model is complex if a resistive character of $Z(f)$ characteristic is not recognized in the all-considered frequency range. Excluding the typical method based on the analytical model described in the domain of complex numbers, resistors may be parameterized by fitting the simulated $Z(f)$ characteristic to the measured one if the values of inductors and capacitors are calculated previously. However, modification of model topology by implementing additional resistors modifies a quality factor at resonant frequencies when capacitors and inductors are parameterized in previous steps. Still, it also may shift resonant frequencies in $Z(f)$ characteristics. Consequently, fitting resistor parameters often requires the readjustment of all the model parameters.

Several optimization and curve-fitting methods have been widely employed for impedance model identification, including the Levenberg-Marquardt algorithm, least-squares approach, and Nelder-Mead simplex method. These traditional methods have demonstrated effectiveness in many applications but often face limitations when dealing with complex impedance profiles exhibiting multiple resonances, as they can converge with local minima or produce biased parameter estimates under noisy conditions. To address these challenges, we adopt the DE algorithm, which has been demonstrated to provide higher robustness and accuracy in parameter extraction for non-linear systems, offering global optimization capabilities that enhance convergence reliability and solution quality.

Impedance Model Formulation and Implementation in Python

Method application requires defining a symbolic expression describing the module of model impedance as a function of frequency. The parameter extraction method has been applied in the Python 3.10 computing language using $lmfit$ and, $numpy$ libraries. The model is prepared and defined as a Python function in the first step. As the inputs of the process, the list of model parameters and the impedance module in the frequency domain are used. The impedance response is defined as two vectors, the frequency and the impedance module, measured using an impedance analyzer. The class parameter should define the model parameters' names, along with their minimum and maximum values that should be limited. If possible, an initial value could be provided if it can be pre-estimated. The basic electrical connection between the model's components is defined for serial and parallel as the Python function:

$defserial\,(Z1, Z2):$
$return\; Z1 + Z2$
$def\,parallel\,(Z1, Z2):$
$return\,(Z1 * Z2)\,/\,(Z1 + Z2)$

where *Z1* and *Z2* are the impedances of the chosen components, with the fitting option constraint that they must always be non-zero. To illustrate the model and the dependencies between the components, the example depicted in Fig. 3 was used. The resistor R_1 and the capacitor C_0 are connected in parallel, and next, their equivalent impedance is in a serial connection with R_0 and L_0. Finally, the total impedance is visible from nodes a_1 and a_2 is defined as

$$Z = serial\,(serial\,(parallel\,(R1, XC0)\,, R0)\,, XL0)$$

where: $X_{L0} = \omega * L_0 * j$ and $X_{C0} = \frac{1}{(\omega * C_0 * j)}$ are the reactance of L_0 and C_0 components, the angular frequency is defined as $\omega = 2 * \pi * f$ (rad/s), where f is the frequency in hertz, π is the mathematical constant, and j denotes the imaginary unit of a complex number.

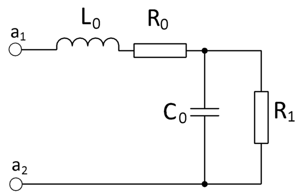


Figure 3: A simple circuit for illustrating model preparation as a Python function.

It is possible to define any circuit containing passive elements using this notation. In some cases, it is necessary to use a delta-star or star-delta transformation. The unsolved problem may be in the case of a circuit where magnetic coupling exists, and the currents in these elements are different. In Python, it is possible to use loops to generate ladder models. This is often done in long transmission lines with duplicated topologically identical cells. First, the topology of the basic cell is defined as in the example above and next, using the loop for i in $range(0, ord)$: Cells are generated. Where ord is the number of the cells. A similar procedure generates the class parameter, including all the ladder parameters.

The proposed extraction method is applied using the $lmfit$ [1] library, which provides curve-fitting computational methods for optimizing a model's fit to data in Python. It provides a convenient front end for tackling nonlinear enhancement and curve-fitting challenges. These methods are critical in many scientific and engineering fields where data-driven insights are critical. SciPy, a popular Python library for implementing these methods, provides several optimizations and curve-fitting functions, including "curve_fit" and "least_sq". Adjusting the model's parameters minimizes the difference between observed data and model predictions. This procedure yields a curve that best represents the underlying trend or pattern in the data, allowing for improved comprehension, prediction, and analysis. The parameter extraction procedure is invoked by using the command

$$fitted\_params = minimize(Zfit, params,$$
$$args = (fr, Zfr,)\,,$$
$$method = \text{`}differential\_evolution\text{'})$$

where $Zfit$ is the objective function, including the impedance model parameters is the class contains a list of model parameters and their bound values, f_r, Zf_r measured the frequency, and the module of impedance vectors. The iterative procedure calculates the model parameters by minimizing the relative difference between the input impedance and a computed impedance using the DE algorithm. As an output result, a class of parameters containing the matched values of the extracted elements is obtained. The extraction process of model parameters is complete when the maximum discrepancy between calculated and experimental impedance values for selected points of characteristic $Z(f)$ is less than the specified error. Then obtained values of model parameters are saved as final procedure results. Calculations were performed using the default settings of the differential evolution algorithm implemented in the lmfit Python library.

- Population size: 15× number of fitting parameters.
- Number of generations: 1000.
- Crossover rate (CR): 0.9 (90%).

- Scaling factor (F): 0.85.
- Mutation strategy: 'best1bin'.

The number of function evaluations depended on the complexity of the tested model and the number of fitting parameters. The total number of function evaluations was determined by the product of the population size and the number of generations and therefore increased with model complexity.

The 'best1bin' strategy in differential evolution selects the best individual and combines it with the difference between two random individuals to create a new candidate solution for optimization according to Eq. (3). The number of function evaluations varied depending on the model complexity and the number of fitting parameters.

4 Simulation Verification

In order to test the accuracy and effectiveness of the DE algorithm in parameter extraction, a circuit simulation was conducted. The simulation data used a known model parameter to determine the method's accuracy. The Saber Sketch from Synopsis was used as the circuit simulator, and the impedance response was obtained using AC signal analysis. The simulation has been performed with logarithmic incrementation of the calculation step in the range from 1 to 20 MHz. The tested circuit was fed from a voltage source, and impedance was measured as the ratio between its voltage and current.

First of all, a simulation was conducted on the basic topology of the Line Impedance Stabilization Network (LISN) as shown in Fig. 4. The obtained $Z(f)$ characteristic from the simulation was taken as the input for the DE extraction method, viewed from the p and n terminals. The resulting output was a list of parameter values, which is presented in Table 1.

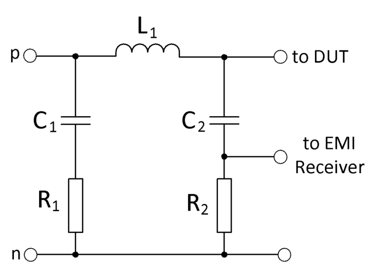


Figure 4: Schematic of a simple LISN used to validate the extraction method based on simulation.

Table 1: Extraction results for LISN.

Parameter	Simulation	DE Algorithm	Error [%]
R_1	5 Ω	4.999 Ω	0.01
R_2	1000 Ω	999.529 Ω	0.04
L_1	50 μH	50.834 μH	1.6
C_1	8 μF	7.999 μF	0.02
C_2	250 nF	250.235 nF	0.09

A comparison of data from the simulations and DE implementation was carried out in Python 3.11 running on Windows 10 Pro (64-bit). A comparison of data from the simulation model and those obtained with DE is shown in Table 1. The test was performed using a Dell Vostro 3560 computer equipped with an Intel(R) Core (TM) i7-3632QM CPU @ 2.20 GHz and 8.00 GB RAM. Computations were performed in serial mode, and switching to parallel computation mode did not significantly affect the quality or duration of the calculations. The input data for the DE algorithm consisted of the impedance characteristic Z[Ω] = f(f[Hz]) measured with a KEYSIGHT E4990A impedance analyzer in the form of an array, while the output data was a vector containing the subsequent values of the parameters R[Ω], L[H], C[F].

Fig. 5 shows the impedance response from the simulation and those generated using the parameter values from the DE algorithm. Since the topology of the model used in the DE algorithm is identical to the simulation model, the method is very well-conditioned, and both graphs coincide almost perfectly.

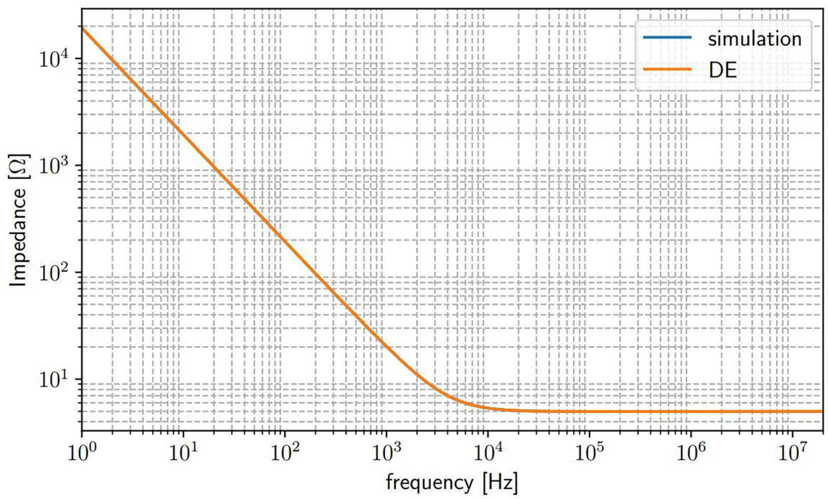


Figure 5: LISN impedance response. Comparison of simulation results and those obtained from DE.

The next example of the DE extraction method using simulation has been performed for the model of the common-mode impedance of induction motor introduced in [6] (Fig. 6).

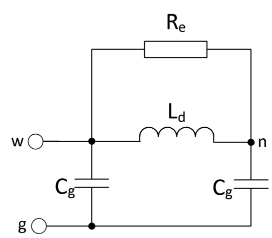


Figure 6: Schematic of the common-mode impedance of induction motor used to validate the extraction method based on simulation.

In this case, the identification results are error-free, which means that the method is effective when the accurate model of the tested object is known. It has been verified that the DE algorithm functions correctly even with more complex circuits that have more elements. Still, some disadvantages may arise when the exact model is not fully known, as with natural objects. Furthermore, comparative results are simulated as shown in Fig. 7. Between impedance and frequency among simulation and DE whereas the extraction results can been seen in Table 2.

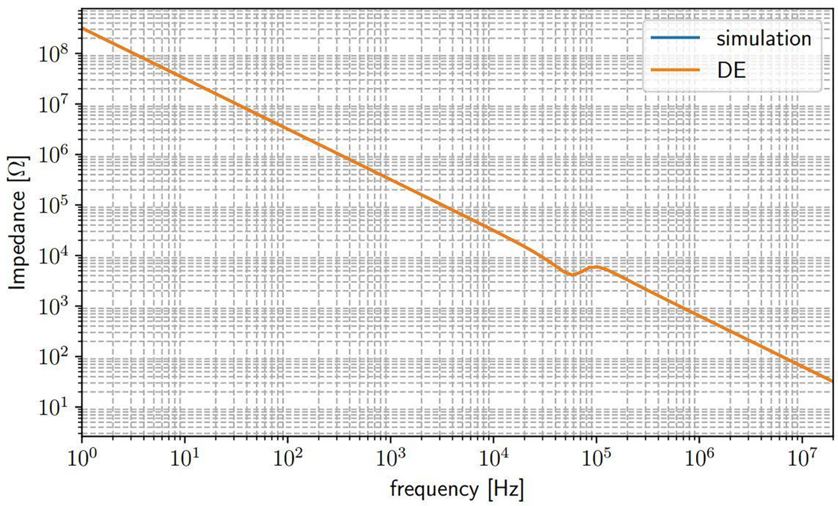


Figure 7: The common-mode impedance of induction motor characteristics. Comparison of simulation results and those obtained from DE.

Table 2: Extraction results for the induction motor.

Parameter	Simulation	DE Algorithm	Error [%]
R_e	17.4 kΩ	17.49 kΩ	0.5
L_d	28.12 mH	28.119 mH	0.003
C_g	0.250 nF	0.2499 nF	0.04

5 Verification of the Extraction Method Using Measurement Data

This section presents the extraction of impedance parameters using DE from data obtained in a laboratory test of natural objects. The measurement of impedance response has been performed using the impedance analyzer KEYSIGHT E4990A (20 Hz–20 MHz). The impedance analyzer was connected to the highlighted terminals of the tested object, and the measurement was carried out by sweeping the frequency with logarithmic increments. Note that, in this article, the authors do not focus on the physical interpretation of the models but mainly on identifying their parameters using the DE algorithm.

5.1 EMI Filter

The first example of using DE was the common mode impedance of the EMI filter. The equivalent circuit of the investigated impedance is shown in Fig. 8. The circuit elements were selected based on the technical data of the filter and complemented by the elements described in Section 3. The circuit includes coil inductances, wire resistances, ground capacitances, and residual resistances.

A comparison between the measured impedance responses and the calculated impedance responses using DE identification parameters is presented in Fig. 9. The calculation time ranged from 3 to 5 s. It can be seen that the two characteristics coincide, although there are slight deviations in the frequency range from 20 to 100 kHz. This is because the filter resistance model is not precisely specified, and additional resistance should probably be added. The accuracy of the presented method can be described by the relative mean error of 0.03% and the relative maximum error of 22.52% between impedance from measurement and the DE algorithm.

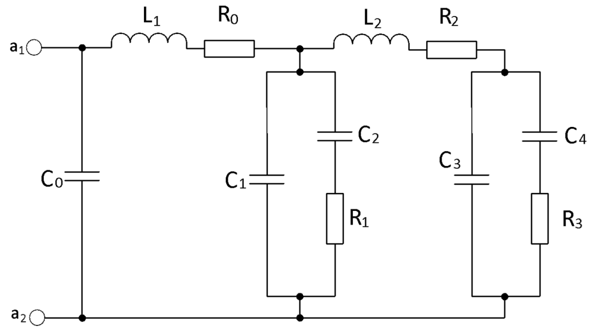


Figure 8: Equivalent circuit of the common-mode impedance of the filter used to validate the extraction method based on measurement.

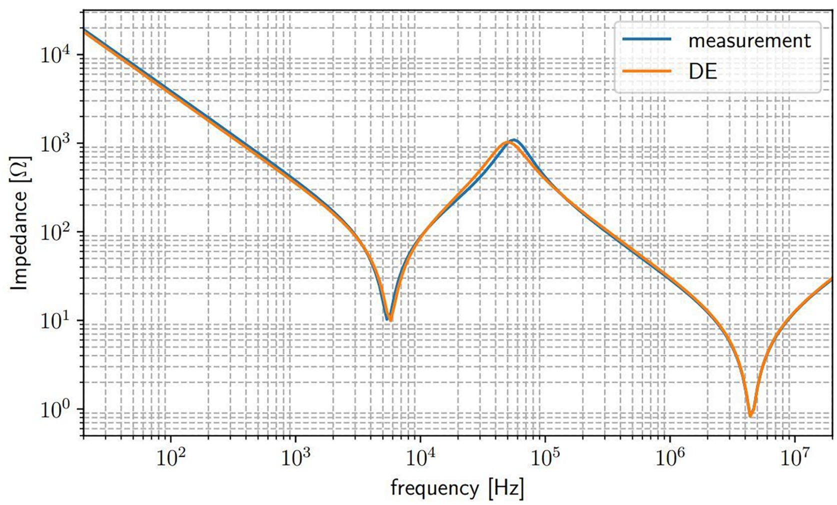


Figure 9: The common-mode impedance of induction motor characteristics. Comparison of measurement results and those obtained from DE.

5.2 The Photovoltaic Panels and Cable

The following example used for parameter extraction is the differential impedance of a string of photovoltaic panels together with a cable (~20 m long), which was disconnected from the terminals of a voltage inverter. The measurement was carried out at night, without the presence of sunlight. In this case, the presented model in Fig. 10, which is a ladder model, can be defined using an iterative procedure (Python loop "for"), with three RLC cells and one input capacitance C_0.

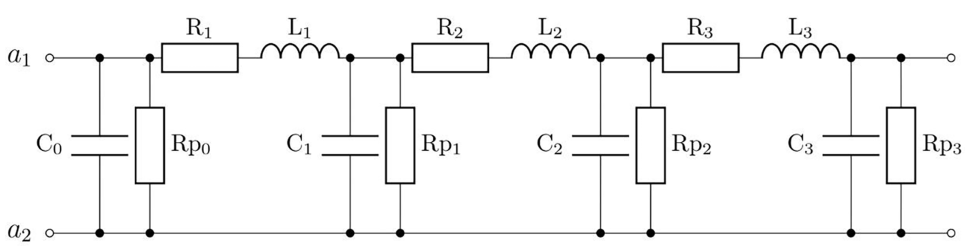


Figure 10: Equivalent circuit of the differential-mode impedance of the photovoltaic string and the cable used to validate the extraction method based on measurement.

After parameter extraction using the DE algorithm, the impedance response has been calculated using the model from Fig. 10 and compared with the measured one in Fig. 11. In this case, both parts have a similar shape, but there are some differences: the level of the resonant peak at 150 kHz, no resonance at 270 kHz, and a different slope at the end of the range. However, these differences result from the fact that the model is not well-conditioned and should be extended with additional components. For example, in the case without parallel resistors Rs, the relative maximum error is more than twice as large. After adding the parallel resistors, the max error decreased 2 times. The computed errors are as follows:

- relative mean error = 0.01%
- relative maximum error = 29.2%

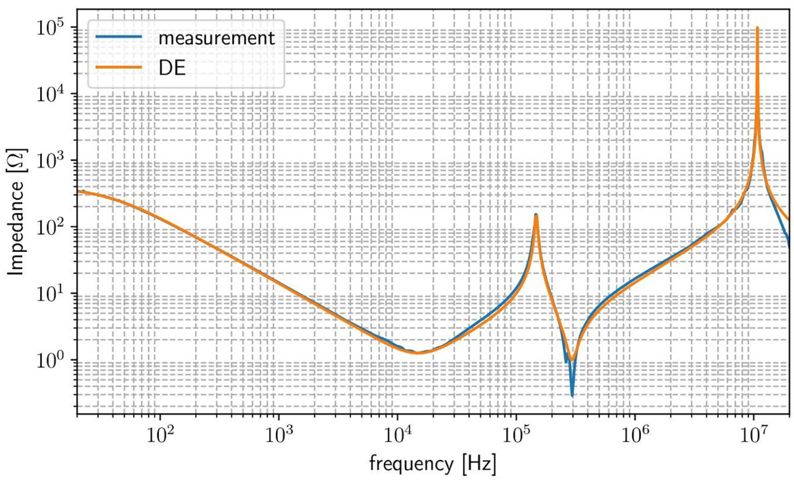


Figure 11: The differential-mode impedance of the photovoltaic string and the cable characteristics. Comparison of measurement results and those obtained from DE.

5.3 The Induction Machines

The equivalent circuit of the differential-mode impedance of the induction machine has been described in [17]. The model topology, presented in Fig. 12, was prepared using the procedure based on the resonant frequencies' identification described in Section 3, and all RLC component values were determined using the DE algorithm.

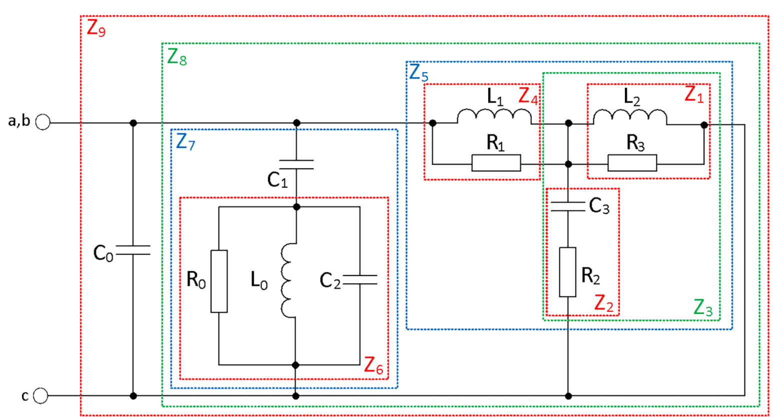


Figure 12: Equivalent circuit of the differential-mode impedance of the induction machine used to validate the extraction method based on measurement.

A comparison between simulated and experimental characteristics $Z(f)$ is presented in Fig. 13. The strong correlation observed between the simulated and experimental $Z(f)$ characteristics for each analyzed object is evident. Generally, the shape of $Z(f)$ elements recovered using parameters obtained by applying DE, resonant frequencies, and impedance values of models near resonant frequencies is close to the experimental results. Some differences result from the model's imperfections, and what should be noticed: resonances around 10 MHz have not been identified at all. This is not due to weakness in the presented method used but to the fact that the applied model is not complete and does not contain the components responsible for this resonance. The computed errors are as follows:

- relative mean error = 0.01%
- relative maximum error = 37.45%

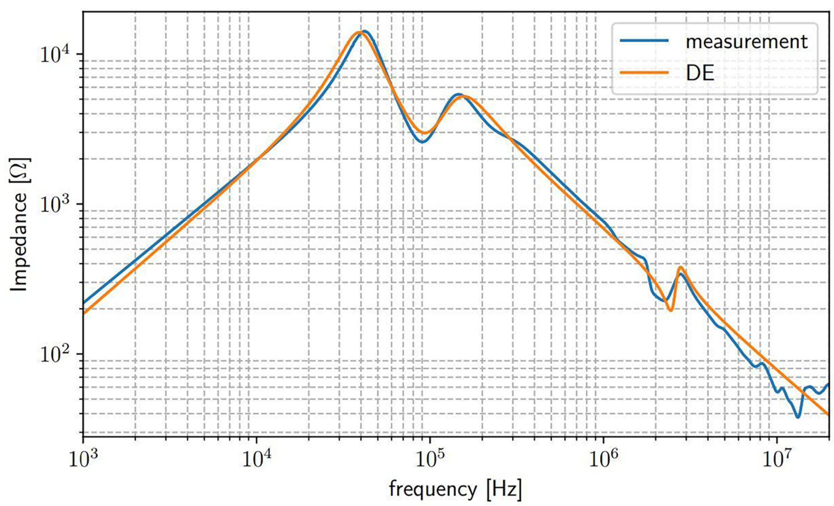


Figure 13: The differential-mode impedance of the induction machine characteristics. Comparison of measurement results and those obtained from DE.

5.4 *The Induction Coils*

The ladder model presented in Fig. 14 was employed to model the impedance response. Due to numerous resonances, a model consisting of five serial ladder steps and one parallel module was used.

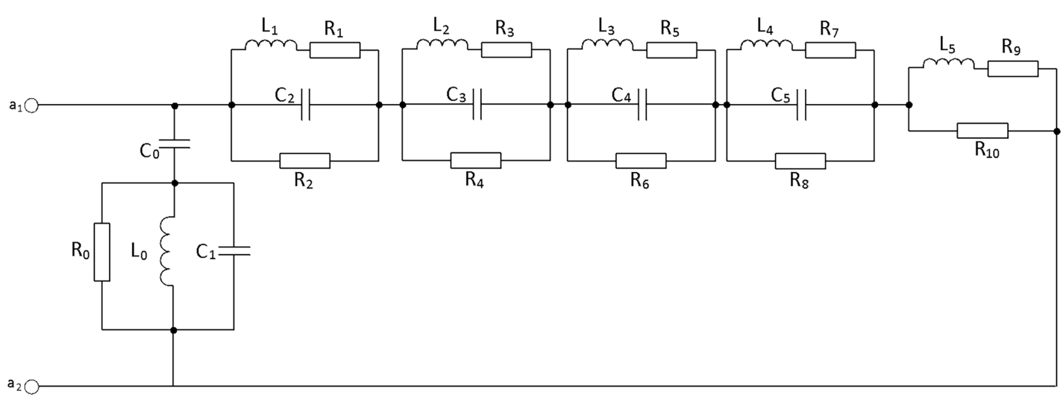


Figure 14: Equivalent circuit of the impedance of the inductor used to validate the extraction method based on measurement.

The result of using the DE algorithm compared to measured data is presented in Fig. 15. The parameter extraction process was highly effective in this case, demonstrating precise reconstruction of both the shape and resonance peaks. This is notable given the complexity of this particular model, surpassing those presented previously. Only for frequencies around 150 kHz, slight differences can be noticed due to the inaccurate nature of the resistive part of the model. The computed errors are as follows:

- relative mean error = 0.01%
- relative maximum error = 50.58%

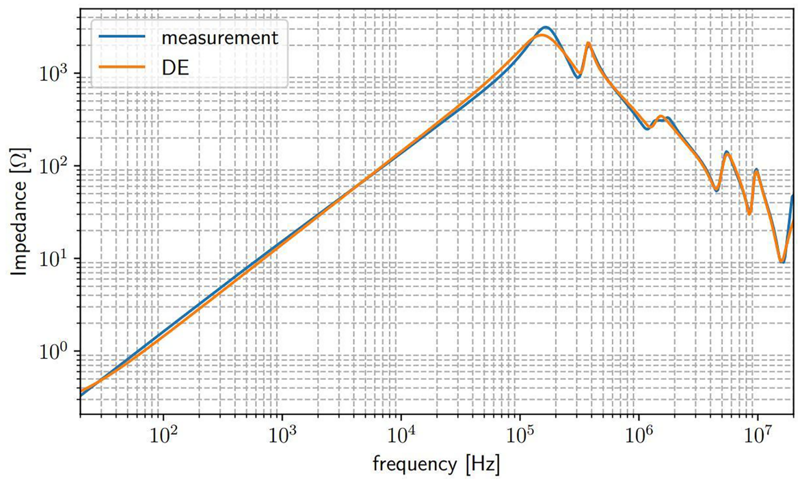


Figure 15: The impedance of the inductor characteristics. Comparison of measurement results and those obtained from DE.

6 Comparison with Existing Optimization Methods

To provide a meaningful comparison between optimization methods, the induction coil equivalent circuit shown in Fig. 14 was selected as the benchmark case. This model represents the most complex circuit investigated in this study, containing multiple resonant peaks and strong parameter interactions. As a result, it provides the most challenging scenario for parameter extraction and enables a rigorous evaluation of the performance of different optimization techniques. Simpler circuit examples were not included in this comparison because their lower complexity produces only minor differences among optimization methods and therefore offers less discrimination of algorithm performance. We conducted a systematic comparison between DE and several well-established optimization methods, including Levenberg–Marquardt, least-squares, Nelder–Mead, Powell, and Cobyla. The results are summarized in Table 3 and illustrated in Fig. 16.

Table 3: Comparison of the DE with existing optimization methods.

Method	Mean Error [%]	Maximal Error [%]
Levenberg-Marquardt	0.04	64.3
Least-Squares	0.06	67.1
Nelder-Mead	0.04	61.6
Powell	0.03	55.3
Cobyla	0.02	56.2
DE	0.01	58.6

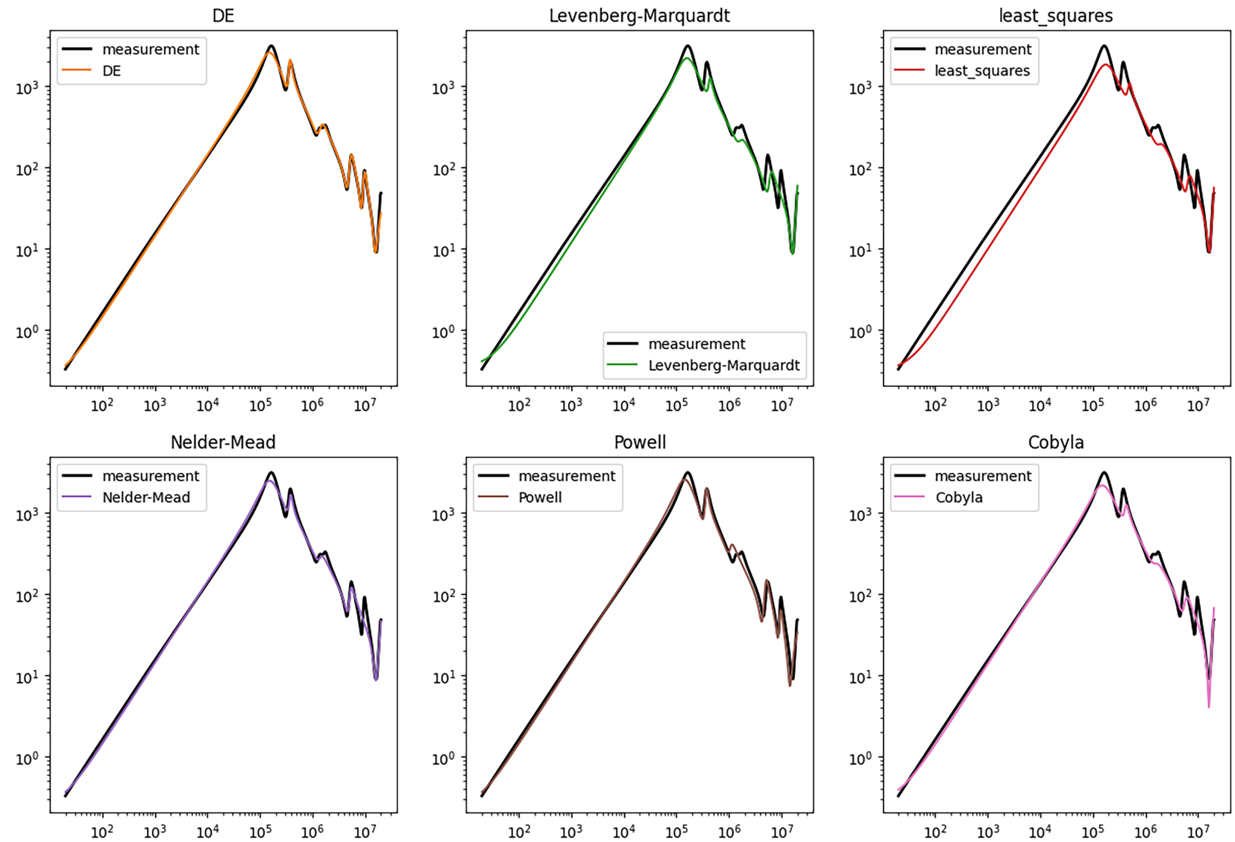


Figure 16: Comparison of the DE with existing optimization methods.

As shown in the table, the mean error across methods ranged from 0.01% to 0.06%, with DE achieving the lowest value (0.01%). Also, the table shows the error metrics for the compared methods appear similar. However, a visual analysis of the figures demonstrates that only the DE-based method correctly identified all resonant peaks. The maximal error values were of a similar order, with DE again providing one of the most favorable results (58.6%). More importantly, the graphical comparison demonstrates that only the DE-based method was able to reproduce all resonant peaks across the frequency spectrum, particularly in circuits with multiple, closely spaced resonances. This comparison confirms that DE delivers the highest accuracy in impedance modeling, which is crucial for reliable EMC analysis of modern power electronic converters.

7 Conclusion

This study presents a robust and scalable approach for extracting the parameters of complex impedance models using the Differential Evolution (DE) algorithm. The proposed method enables accurate identification of RLC component values in equivalent circuits, including systems characterized by multiple resonances and strong parameter interdependencies. Implemented in Python using open-source scientific libraries, the approach is easily reproducible and can be readily integrated into simulation, modeling, and optimization workflows.

The method was validated using both simulated and experimentally measured impedance responses obtained from a range of practical electrical systems. The results demonstrate that DE can accurately reproduce impedance characteristics over a wide frequency range and effectively identify model parameters, provided that the selected circuit topology adequately represents the physical system. Furthermore, the method supports complex equivalent circuit structures without imposing restrictions on the number of components, making it suitable for advanced energy systems, impedance-based diagnostics, and electromagnetic compatibility studies.

Comparative analysis with several established optimization techniques showed that DE provides superior performance in identifying resonant behaviors and achieving accurate parameter estimation for complex impedance models. Accurate extraction of operational and parasitic parameters can support early-stage EMI prediction, converter design optimization, PCB layout improvement, filter design, cable selection, and the development of digital twins for power conversion systems. In addition, the proposed methodology offers potential for condition monitoring and aging assessment, where variations in extracted parameters may serve as indicators of component degradation and system health.

Acknowledgement: The authors express gratitude to Gdańsk University of Technology for providing lab facility and acknowledge the equal research contributions from the Research Unit of Robophilosophy and Integrative Social Robotics (RISR), Aarhus University, Denmark.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Piotr Musznicki contributed to the conceptualization and methodological development of the study, supervised the research, and participated extensively in reviewing and editing the manuscript. Marek Turzyński led the software development, validation, and formal analysis and made a major contribution to preparing the original draft. Ghulam E Mustafa Abro contributed substantially to the conceptualization and methodology, coordinated the project administration, and participated in the critical review and editing of the manuscript. Viola Gierszewska contributed to software development, validation, and the provision of research resources. Lyu Guanghua contributed to the investigation, data curation, and visualization. Arsalan Muhammad Soomar contributed to the formal analysis, visualization, and preparation of the original draft. Syed Hadi Hussain Shah contributed to the investigation, data curation, and validation. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The source code used in this study is part of an ongoing research and development project and is currently undergoing further validation and institutional review. The code will be made publicly available through an online repository upon completion of the project and the necessary approval procedures. All methodological details, model configurations, and experimental settings are provided in the manuscript to facilitate independent reproduction of the reported results.

Ethics Approval: Not applicable. This study did not involve human or animal subjects.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Newville M, Stensitzki T, Allen DB, Ingargiola A. LMFIT: non-linear least-square minimization and curve-fitting for Python. Zenodo. 2014:11813. doi:10.5281/zenodo.11813.
2. Grandi G, Casadei D, Reggiani U. Equivalent circuit of mush wound AC windings for high frequency analysis. In: Proceedings of the ISIE' 97 Proceeding of the IEEE International Symposium on Industrial Electronics; 1997 Jul 7–11; Guimaraes, Portugal. p. SS201–6. doi:10.1109/ISIE.1997.651762.
3. Miloudi H, Bendaoud A, Benhadjla M, Gourbi A. Modeling of differential-mode and common-mode characteristics for EMI/EMC analysis applied to a high-frequency induction motor. Acta Electrotehn. 2008;49(3):328–33.
4. Stevanovic I, Skibin S, Masti M, Laitinen M. Behavioral modeling of chokes for EMI simulations in power electronics. IEEE Trans Power Electron. 2013;28(2):695–705. doi:10.1109/TPEL.2012.2203319.
5. Naishadham K. Closed-form design formulas for the equivalent circuit characterization of ferrite inductors. IEEE Trans Electromagn Compat. 2011;53(4):923–32. doi:10.1109/TEMC.2011.2116795.

6. Boglietti A, Carpaneto E. Induction motor high-frequency model. In: Proceedings of the Conference Record of the 1999 IEEE Industry Applications Conference. Thirty-Forth IAS Annual Meeting (Cat. No.99CH36370); 1999 Oct 3–7; Phoenix, AZ, USA. p. 1551–8.
7. Gong X, Ferreira JA. Extracting the parameters of a common-mode EMI equivalent circuit model for a drive inverter. In: Proceedings of the 2010 International Power Electronics Conference-ECCE ASIA; 2010 Jun 21–24. p. 892–9.
8. Bade TG, Roudet J, Guichon JM, Sartori CAF, Kuo-Peng P, Schanen JL, et al. Frequency-domain modeling of unshielded multiconductor power cables for periodic excitation with new experimental protocol for wide band parameter identification. Electr Eng. 2019;101(2):333–43. doi:10.1007/s00202-019-00780-2.
9. Janeiro FM, Ramos PM. Application of genetic algorithms for estimation of impedance parameters of two-terminal networks. In: Proceedings of the 2009 IEEE Instrumentation and Measurement Technology Conference; 2009 May 5–7; Singapore. p. 602–6. doi:10.1109/IMTC.2009.5168521.
10. Turzyński M. Induction machine behavioral modeling for prediction of EMI propagation. Bull Pol Acad Sci Tech Sci. 2017;65(2):247–54. doi:10.1515/bpasts-2017-0028.
11. Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59. doi:10.1023/a:1008202821328.
12. Farda I, Thammano A. An adaptive differential evolution with multiple crossover strategies for optimization problems. HighTech Innov J. 2024;5(2):231–58. doi:10.28991/hij-2024-05-02-02.
13. Ahmad MF, Isa NAM, Lim WH, Ang KM. Differential evolution: a recent review based on state-of-the-art works. Alex Eng J. 2022;61(5):3831–72. doi:10.1016/j.aej.2021.09.013.
14. Zitzler E. Evolutionary multiobjective optimization. In: Handbook of natural computing. Berlin/Heidelberg, Germany: Springer; 2012. p. 871–904. doi:10.1007/978-3-540-92910-9_28.
15. Bilal, Pant M, Zaheer H, Garcia-Hernandez L, Abraham A. Differential evolution: a review of more than two decades of research. Eng Appl Artif Intell. 2020;90(1):103479. doi:10.1016/j.engappai.2020.103479.
16. Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Computat. 2009;13(2):398–417. doi:10.1109/tevc.2008.927706.
17. Zhu W, Tang Y, Fang JA, Zhang W. Adaptive population tuning scheme for differential evolution. Inf Sci. 2013;223(3):164–91. doi:10.1016/j.ins.2012.09.019.
18. Huang Z, Chen Y. An improved differential evolution algorithm based on adaptive parameter. J Control Sci Eng. 2013;2013(4):462706. doi:10.1155/2013/462706.
19. Zheng LM, Zhang SX, Tang KS, Zheng SY. Differential evolution powered by collective information. Inf Sci. 2017;399(3):13–29. doi:10.1016/j.ins.2017.02.055.
20. Aljaidi M, Agrawal SP, Jangir P, Pandya SB, Parmar A, Arpita, et al. A two stage differential evolution algorithm for parameter estimation of proton exchange membrane fuel cell. Sci Rep. 2025;15(1):5354. doi:10.1038/s41598-025-89304-6.
21. Hermite C, Douiri MR. Differential evolution algorithm featuring novel mutation combined with Newton-Raphson method for enhanced photovoltaic parameter extraction. Energy Convers Manag. 2025;326:119468. doi:10.1016/j.enconman.2024.119468.
22. Wu H. Analysis and design of insertion loss of EMI filter based on improved differential evolution algorithm. In: Proceedings of the 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC); 2023 Jun 16–17; Dharwad, India. p. 1–5. doi:10.1109/icaisc58445.2023.10199883.
23. Mueller D, Beltle M, Tenbohlen S. Automated filter optimization for high-voltage cable harness based on circuit simulations for conducted emissions prediction. In: Proceedings of the 2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE); 2018 Aug 27–30; Amsterdam, the Netherlands. p. 84–9. doi:10.1109/EMCEurope.2018.8485112.

24. Xu D, Song X, Wu Z, Xu J, Hu Q. A Levenberg-Marquardt algorithm-based line parameters identification method for distribution network considering multisource measurement. IET Renew Power Gener. 2024;18(16):3743–52. doi:10.1049/rpg2.12955.
25. González F, Greiner D, Mena V, Souto RM, Santana JJ, Aznárez JJ. Fitting procedure based on differential evolution to evaluate impedance parameters of metal–coating systems. Eng Comput. 2019;36(9):2960–82. doi:10.1108/ec-11-2018-0513.