Index-aware learning of circuits

Idoia Cortes Garcia[1] | Peter Förster*[1,2] | Lennart Jansen[2] | Wil Schilders[1] | Sebastian Schöps[2]

[1]Eindhoven University of Technology, Netherlands

[2]Technical University of Darmstadt, Germany

Correspondence
*Peter Förster, Eindhoven University of Technology. Email: p.f.forster@tue.nl

Summary

Electrical circuits are present in a variety of technologies, making their design an important part of computer aided engineering. The growing number of tunable parameters that affect the final design leads to a need for new approaches of quantifying their impact. Machine learning may play a key role in this regard, however current approaches often make suboptimal use of existing knowledge about the system at hand. In terms of circuits, their description via modified nodal analysis is well-understood. This particular formulation leads to systems of differential-algebraic equations (DAEs) which bring with them a number of peculiarities, e.g. hidden constraints that the solution needs to fulfill. We aim to use the recently introduced dissection concept for DAEs that can decouple a given system into ordinary differential equations, only depending on differential variables, and purely algebraic equations that describe the relations between differential and algebraic variables. The idea then is to only learn the differential variables and reconstruct the algebraic ones using the relations from the decoupling. This approach guarantees that the algebraic constraints are fulfilled up to the accuracy of the nonlinear system solver, which represents the main benefit highlighted in this article.

KEYWORDS:
electrical circuits, differential-algebraic equations, modified nodal analysis, dissection concept, machine learning

1 | INTRODUCTION

Design optimization and uncertainty quantification are ubiquitous tools of modern computer aided engineering. Due to the ever-increasing complexity of engineering systems, machine learning techniques have gained popularity for constructing surrogate models when the objective function becomes expensive to evaluate and a large number of (design or uncertainty) parameters are present. In these cases, classical model order reduction techniques[1] or function approximation approaches[2] suffer from the curse of dimensionality: the number of operations to construct and the memory required to store the model grow exponentially with respect to the number of parameters. Experimental and in some cases even theoretical evidence[3] shows that machine learning approaches may be able to overcome the curse of dimensionality and provide surrogates that are fast to evaluate, while requiring comparably little data.

In the particular case of electrical circuit design, neural networks have been used for design optimization for over 20 years[4,5]. More recently, Gaussian process regression has been employed for both uncertainty quantification and design optimization during analog integrated circuit design[6,7]. The commonality between these approaches is that they focus on the learning part: they aim to provide a computationally efficient and accurate surrogate, given data produced by some circuit simulation software.

Thus, they all treat the circuit simulator as a black box that simply provides the data which is then used for constructing the surrogate. In contrast, we want to exploit the known structure that underlies the equations describing electrical circuits.

More specifically, we consider the modified nodal analysis[8] (MNA). MNA is one of the most popular circuit descriptions and lies at the center of SPICE-like simulation software such as LTspice[9], Xyce[10] and PSpice[11]. Applying MNA to a given circuit leads to systems of differential-algebraic equations (DAEs), which can generally be written as systems of implicit differential equations[12]

$$\mathbf{f}(\mathbf{x}', \mathbf{x}, t, \mathbf{p}) = \mathbf{0}, \quad \mathbf{x}(0) = \mathbf{x}_0,$$

where the Jacobian $\mathbf{J}_{\mathbf{x}'}(\mathbf{f})$, of $\mathbf{f}$ w.r.t. $\mathbf{x}'$, is singular and $\mathbf{p}$ are the design or uncertainty parameters. To give a more intuitive description, one can think of DAEs as ordinary differential equations (ODEs) that are constrained to a manifold determined by (hidden) constraints on the solution variables. Multiple so-called index concepts were introduced in order to characterize the difficulties that appear in the context of DAEs. A reference by Mehrmann[13] outlines some of the most prominent, including the differentiation, tractability, strangeness and perturbation indices. For all of these, the index $\nu \in \mathbb{N}$ depends on structural properties of the DAE and (informally) measures the degree of difficulty associated with solving it. In this sense, higher index DAEs usually lead to more problems, both analytically and numerically.

Various relationships between the indices, including situations when they coincide, are also known[13] and the structure and index of MNA are well understood for both independent sources[14] and controlled sources[15]. Rather recently, the dissection concept was introduced[16]. It combines features of the strangeness and tractability indices to decouple a DAE into an ODE and a set of purely algebraic equations. In particular, the entire dynamics of the solution may then be found only using the ODE. We aim to use this dissection concept to propose an approach for learning electrical circuits more accurately and efficiently.

In the following, section 2 introduces MNA in more detail and states some well-known related results. Afterwards, section 3 outlines the dissection concept and showcases its properties using small example circuits. The same examples are then used to present the new approach on an abstract level in section 4 and on a numerical level in section 5, where we also showcase an alternative method that trades some of the accuracy of our approach for an easier implementation. Preliminary conclusions about the effectiveness of our approach and future research directions are given in section 6.

2 | MODIFIED NODAL ANALYSIS

We begin by looking at the system of DAEs that results from MNA. From here onward we will disregard controlled sources. As these are of crucial importance for engineering applications however, we will state whether extensions including controlled sources are available or missing at the appropriate times. Borrowing the notation from Tischendorf[14], and only explicitly expressing the time dependence of independent sources for brevity, the system of MNA reads

$$\mathbf{A}_C \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{q}_C(\mathbf{A}_C^\top \boldsymbol{\varphi}) + \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \boldsymbol{\varphi}) + \mathbf{A}_L \mathbf{i}_L + \mathbf{A}_V \mathbf{i}_V + \mathbf{A}_I \mathbf{i}_s(t) = \mathbf{0} \tag{1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\phi}_L(\mathbf{i}_L) - \mathbf{A}_L^\top \boldsymbol{\varphi} = \mathbf{0} \tag{1b}$$

$$\mathbf{A}_V^\top \boldsymbol{\varphi} - \mathbf{v}_s(t) = \mathbf{0}, \tag{1c}$$

where $\mathbf{g}_R$, $\mathbf{q}_C$ and $\boldsymbol{\phi}_L$ represent functions modeling resistive, capacitive or inductive devices respectively. The terms for independent current and voltage sources are given by $\mathbf{i}_s(t)$ and $\mathbf{v}_s(t)$, while $\boldsymbol{\varphi}$ denotes the vector of nodal potentials and $\mathbf{i}_L$, $\mathbf{i}_V$ are the currents flowing through branches containing inductors or voltage sources respectively. The last ingredient is given by the incidence matrices $\mathbf{A}_*$, where $*$ indicates the device type. These collect the branch to node relations of the underlying electrical network when considering the branches and nodes as edges and vertices of a graph. In the context of design optimization or uncertainty quantification, the device functions $\mathbf{g}_R$, $\mathbf{q}_C$ or $\boldsymbol{\phi}_L$ could e.g. depend on (subsets of) the parameters $\mathbf{p}$.

In order to obtain a version of (1) that is better suited to analysis and implementation, we consider the device function Jacobians

$$\mathbf{L}(\mathbf{i}_L) := \mathbf{J}_{\mathbf{i}_L}(\boldsymbol{\phi}_L), \quad \mathbf{C}(\mathbf{A}_C^\top \boldsymbol{\varphi}) := \mathbf{J}_{\mathbf{A}_C^\top \boldsymbol{\varphi}}(\mathbf{q}_C), \tag{2}$$

where we use the same notation for the Jacobians as in section 1. We also assume that the device function for resistors may be expressed as

$$\mathbf{g}_R(\mathbf{A}_R^\top \boldsymbol{\varphi}) = \mathbf{G}(\mathbf{A}_R^\top \boldsymbol{\varphi}) \mathbf{A}_R^\top \boldsymbol{\varphi}. \tag{3}$$

In practice, this is accomplished by either considering the chordal (static) conductance $\frac{[\mathbf{g}_R]_i}{[\mathbf{A}_R^\top\boldsymbol{\varphi}]_i}$ for each resistor R_i separately or by using the Jacobian $\mathbf{J}_{\mathbf{A}_R^\top\boldsymbol{\varphi}}(\mathbf{g}_R)$. Inserting (2) and (3) into the original system and writing it in matrix form yields

$$
\begin{bmatrix} \mathbf{A}_C\mathbf{C}(\mathbf{A}_C^\top\boldsymbol{\varphi})\mathbf{A}_C^\top & & \\ & \mathbf{L}(\mathbf{i}_L) & \\ & & \mathbf{0} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \boldsymbol{\varphi} \\ \mathbf{i}_L \\ \mathbf{i}_V \end{bmatrix} + \begin{bmatrix} \mathbf{A}_R\mathbf{G}(\mathbf{A}_R^\top\boldsymbol{\varphi})\mathbf{A}_R^\top & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^\top & & \\ -\mathbf{A}_V^\top & & \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi} \\ \mathbf{i}_L \\ \mathbf{i}_V \end{bmatrix} + \begin{bmatrix} \mathbf{A}_I\mathbf{i}_s(t) \\ \mathbf{0} \\ \mathbf{v}_s(t) \end{bmatrix} = \mathbf{0}, \tag{4}
$$

where we assume that $\mathbf{q}_C$ and $\boldsymbol{\phi}_L$ do not explicitly depend on time. MNA is usually presented in the form of (4) and for it, there exists the following well-known topological index result.

Assuming the matrix-valued functions $\mathbf{G}$, $\mathbf{L}$, $\mathbf{C}$ from (2) and (3) positive definite:

1. The index of (4) is $\nu \leq 2$.

2. The index of (4) is $\nu \leq 1$, if and only if there are no loops consisting only of capacitors and voltage sources containing at least one voltage source and no cutsets consisting only of inductors and current sources.

The result can be found in terms of the tractability index [14], the differentiation index [15] and the dissection concept [16].

We remark that there also exist extensive results about when the index of MNA including controlled sources does not exceed $\nu = 2$ [15].

3 | DISSECTION CONCEPT

This section serves as a general introduction to the dissection concept and more specifically its application to MNA as given in (4). Consider a linear DAE in standard form [16],

$$
\mathbf{M}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} + \mathbf{K}\mathbf{x} + \mathbf{f}(t) = \mathbf{0}, \quad \ker\mathbf{M} \supsetneq \{\mathbf{0}\}, \tag{5}
$$

where we again keep the time dependence of $\mathbf{x}$ implicit. We now demonstrate the first two steps of the dissection concept when applied to systems of the form of (5) and remark that although (4) is a nonlinear DAE, the decoupling still takes the same form in this case due to the special structure of MNA [16].

3.1 | Index one case

Assuming $\mathbf{M} \in \mathbb{R}^{n\times m}$, we define four basis functions $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{V}$, $\mathbf{W}$ such that

$$
\mathrm{im}\,\mathbf{Q} = \ker\mathbf{M}, \quad \mathrm{im}\,\mathbf{W} = \ker\mathbf{M}^\top
$$

and the columns of $\mathbf{P}$ and $\mathbf{Q}$ together form a basis of $\mathbb{R}^n$, while the columns of $\mathbf{V}$ and $\mathbf{W}$ together form a basis of $\mathbb{R}^m$.

Splitting the solution

The key idea is to use the basis functions to split the solution $\mathbf{x}$ into two parts

$$
\mathbf{x} = \mathbf{P}\tilde{\mathbf{x}} + \mathbf{Q}\bar{\mathbf{x}}, \tag{6}
$$

where $\tilde{\ }$ is used to indicate differential (dynamic) variables and $\bar{\ }$ signifies algebraic (fixed) variables. When inserting the splitting (6) into the general system (5) we observe the following

$$
\mathbf{M}\mathbf{P}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}} + \mathbf{K}\mathbf{P}\tilde{\mathbf{x}} + \mathbf{K}\mathbf{Q}\bar{\mathbf{x}} + \mathbf{f}(t) = \mathbf{0} \tag{7}
$$

and this motivates the notation, as only $\tilde{\mathbf{x}}$ appears differentiated in time.

Splitting the system

The procedure then continues by multiplying (7) with $\mathbf{V}^\top$ and $\mathbf{W}^\top$ from the left. This yields

$$\underbrace{\mathbf{V}^\top\mathbf{MP}}_{=:\tilde{\mathbf{M}}}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}} + \underbrace{\mathbf{V}^\top\mathbf{KP}}_{=:\tilde{\mathbf{K}}_{\mathbf{P}}}\tilde{\mathbf{x}} + \underbrace{\mathbf{V}^\top\mathbf{KQ}}_{=:\tilde{\mathbf{K}}_{\mathbf{Q}}}\bar{\mathbf{x}} + \underbrace{\mathbf{V}^\top\mathbf{f}(t)}_{=:\tilde{\mathbf{f}}(t)} = \mathbf{0} \tag{8a}$$

$$\underbrace{\mathbf{W}^\top\mathbf{KP}}_{=:\bar{\mathbf{K}}_{\mathbf{P}}}\tilde{\mathbf{x}} + \underbrace{\mathbf{W}^\top\mathbf{KQ}}_{=:\bar{\mathbf{K}}_{\mathbf{Q}}}\bar{\mathbf{x}} + \underbrace{\mathbf{W}^\top\mathbf{f}(t)}_{=:\bar{\mathbf{f}}(t)} = \mathbf{0}, \tag{8b}$$

where we also introduce shorthands for the arising products. The key observations regarding (8) are the following: $\tilde{\mathbf{M}}$ is regular by construction, thus if $\bar{\mathbf{K}}_{\mathbf{Q}}$ is regular, (8b) gives a purely algebraic equation for $\bar{\mathbf{x}}$ in terms of $\tilde{\mathbf{x}}$. Solving this equation for $\bar{\mathbf{x}}$ and inserting the solution into (8a) then yields an ODE in the differential variables $\tilde{\mathbf{x}}$. In this case, the **(dissection) index** of the DAE is defined to be $\nu = 1$.

3.2 | Index two case

There are of course many DAEs, including those described by MNA, which have a higher index. For these the dissection concept proceeds by introducing additional basis functions and continuing the splitting process in a similar fashion.

Splitting the algebraic variables

We begin by considering basis functions $\bar{\mathbf{P}}, \bar{\mathbf{Q}}, \bar{\mathbf{V}}, \bar{\mathbf{W}}$ of $\bar{\mathbf{K}}_{\mathbf{Q}}$ and split the algebraic variables $\bar{\mathbf{x}}$ further as follows

$$\bar{\mathbf{x}} = \bar{\mathbf{P}}\bar{\mathbf{x}}_{\mathbf{P}} + \bar{\mathbf{Q}}\bar{\mathbf{x}}_{\mathbf{Q}}. \tag{9}$$

Inserting this decomposition into (8b) and multiplying once by $\bar{\mathbf{V}}^\top$ and $\bar{\mathbf{W}}^\top$ from the left yields

$$\bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{x}} + \bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\bar{\mathbf{x}}_{\mathbf{P}} + \bar{\mathbf{V}}^\top\bar{\mathbf{f}}(t) = \mathbf{0} \tag{10a}$$

$$\bar{\mathbf{W}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{x}} + \bar{\mathbf{W}}^\top\bar{\mathbf{f}}(t) = \mathbf{0}. \tag{10b}$$

Splitting the differential variables

We now need basis functions $\tilde{\mathbf{P}}, \tilde{\mathbf{Q}}$ of $\bar{\mathbf{W}}^\top\bar{\mathbf{K}}_{\mathbf{P}}$ to split the differential variables $\tilde{\mathbf{x}}$ as

$$\tilde{\mathbf{x}} = \tilde{\mathbf{P}}\tilde{\mathbf{x}}_{\mathbf{P}} + \tilde{\mathbf{Q}}\tilde{\mathbf{x}}_{\mathbf{Q}} \tag{11}$$

and inserting this splitting into (10b) then allows for an explicit description of $\tilde{\mathbf{x}}_{\mathbf{P}}$

$$\tilde{\mathbf{x}}_{\mathbf{P}} = -\left(\bar{\mathbf{W}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{P}}\right)^{-1}\bar{\mathbf{W}}^\top\bar{\mathbf{f}}(t) =: \tilde{\mathbf{f}}_{\mathbf{P}}(t),$$

since $\bar{\mathbf{W}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{P}}$ is assumed to be regular (otherwise the DAE is underdetermined[16]).

Splitting the solution

Similarly, inserting (11) into (10a) gives a description of $\bar{\mathbf{x}}_{\mathbf{P}}$ in terms of $\tilde{\mathbf{x}}_{\mathbf{Q}}$ as $\bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}$ is guaranteed to be nonsingular

$$\bar{\mathbf{x}}_{\mathbf{P}} = -\left(\bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\right)^{-1}\left(\bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{Q}}\tilde{\mathbf{x}}_{\mathbf{Q}} + \bar{\mathbf{V}}^\top\bar{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{P}}\tilde{\mathbf{x}}_{\mathbf{P}} + \bar{\mathbf{V}}^\top\bar{\mathbf{f}}(t)\right) =: \bar{\mathbf{K}}\tilde{\mathbf{x}}_{\mathbf{Q}} + \bar{\mathbf{f}}_{\mathbf{P}}(t).$$

Expanding $\tilde{\mathbf{x}}$ and $\bar{\mathbf{x}}$ in (8a), according to (9) and (11) respectively, and using the above expressions for $\tilde{\mathbf{x}}_{\mathbf{P}}$ and $\bar{\mathbf{x}}_{\mathbf{P}}$ gives

$$\tilde{\mathbf{M}}\tilde{\mathbf{Q}}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}}_{\mathbf{Q}} + \left(\tilde{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{Q}} + \tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\bar{\mathbf{K}}\right)\tilde{\mathbf{x}}_{\mathbf{Q}} + \tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{Q}}\bar{\mathbf{x}}_{\mathbf{Q}} + \hat{\mathbf{f}}(t) = \mathbf{0}, \tag{12}$$

where we define

$$\hat{\mathbf{f}}(t) := \tilde{\mathbf{M}}\tilde{\mathbf{P}}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{f}}_{\mathbf{P}}(t) + \tilde{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{P}}\tilde{\mathbf{f}}_{\mathbf{P}}(t) + \tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\bar{\mathbf{f}}_{\mathbf{P}}(t) + \tilde{\mathbf{f}}(t).$$

Splitting the system

Finally, we require basis functions $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{W}}$ of $\tilde{\mathbf{M}}\tilde{\mathbf{Q}}$. Multiplying (12) from the left by $\tilde{\mathbf{V}}^{\mathsf{T}}$ and $\tilde{\mathbf{W}}^{\mathsf{T}}$ and redefining $\tilde{\mathbf{x}}_2 := \tilde{\mathbf{x}}_{\mathbf{Q}}$ and $\bar{\mathbf{x}}_2 := \bar{\mathbf{x}}_{\mathbf{Q}}$ yields

$$\underbrace{\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{M}}\tilde{\mathbf{Q}}}_{=:\tilde{\mathbf{M}}_2}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}}_2 + \underbrace{\tilde{\mathbf{V}}^{\mathsf{T}}\left(\tilde{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{Q}} + \tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\bar{\mathbf{K}}\right)}_{=:\tilde{\mathbf{K}}_{\mathbf{P}_2}}\tilde{\mathbf{x}}_2 + \underbrace{\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{Q}}}_{=:\tilde{\mathbf{K}}_{\mathbf{Q}_2}}\bar{\mathbf{x}}_2 + \underbrace{\tilde{\mathbf{V}}^{\mathsf{T}}\hat{\mathbf{f}}(t)}_{=:\tilde{\mathbf{f}}_2(t)} = \mathbf{0} \tag{13a}$$

$$\underbrace{\tilde{\mathbf{W}}^{\mathsf{T}}\left(\tilde{\mathbf{K}}_{\mathbf{P}}\tilde{\mathbf{Q}} + \tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{P}}\bar{\mathbf{K}}\right)}_{=:\bar{\mathbf{K}}_{\mathbf{P}_2}}\tilde{\mathbf{x}}_2 + \underbrace{\tilde{\mathbf{W}}^{\mathsf{T}}\tilde{\mathbf{K}}_{\mathbf{Q}}\bar{\mathbf{Q}}}_{=:\bar{\mathbf{K}}_{\mathbf{Q}_2}}\bar{\mathbf{x}}_2 + \underbrace{\tilde{\mathbf{W}}^{\mathsf{T}}\hat{\mathbf{f}}(t)}_{=:\bar{\mathbf{f}}_2(t)} = \mathbf{0}. \tag{13b}$$

We observe that (13) is of the same form as (8) and that $\tilde{\mathbf{M}}_2$ is again regular by construction. Therefore, if $\bar{\mathbf{K}}_{\mathbf{Q}_2}$ is also regular we can extract an ODE analogous to the previous case by solving (13b) for $\bar{\mathbf{x}}_2$ (in terms of $\tilde{\mathbf{x}}_2$) and inserting that solution into (13a). The DAE is then said to have **dissection index** $\nu = 2$ and we call $\tilde{\mathbf{x}}_2$ the differential variables and $\bar{\mathbf{x}}_2$ the algebraic variables. Furthermore, since (13) and (8) are structurally identical, we may simply apply the second step of the decoupling procedure to (13) in order to deal with even higher index DAEs.

To clarify the naming convention for the following sections, we call $\tilde{\mathbf{x}}_2$ and $\bar{\mathbf{x}}_2$ the purely differential and algebraic variables of an index two DAE respectively, whereas $\bar{\mathbf{x}}$ denotes the explicit (obvious) algebraic variables and $\tilde{\mathbf{x}}$ still contains hidden (implicit) algebraic variables that are identified in the second step of the dissection concept.

3.3 | Dissection concept and MNA

Before we consider example circuits, we take a brief look at the first step of the dissection concept for MNA. For this, we recall the system of MNA (4)

$$\begin{bmatrix} \mathbf{A}_{\mathrm{C}}\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})\mathbf{A}_{\mathrm{C}}^{\mathsf{T}} & & \\ & \mathbf{L}(\mathbf{i}_{\mathrm{L}}) & \\ & & \mathbf{0} \end{bmatrix}\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \boldsymbol{\varphi} \\ \mathbf{i}_{\mathrm{L}} \\ \mathbf{i}_{\mathrm{V}} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{\mathrm{R}}\mathbf{G}(\mathbf{A}_{\mathrm{R}}^{\mathsf{T}}\boldsymbol{\varphi})\mathbf{A}_{\mathrm{R}}^{\mathsf{T}} & \mathbf{A}_{\mathrm{L}} & \mathbf{A}_{\mathrm{V}} \\ -\mathbf{A}_{\mathrm{L}}^{\mathsf{T}} & & \\ -\mathbf{A}_{\mathrm{V}}^{\mathsf{T}} & & \end{bmatrix}\begin{bmatrix} \boldsymbol{\varphi} \\ \mathbf{i}_{\mathrm{L}} \\ \mathbf{i}_{\mathrm{V}} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{\mathrm{I}}\mathbf{i}_{\mathrm{s}}(t) \\ \mathbf{0} \\ \mathbf{v}_{\mathrm{s}}(t) \end{bmatrix} = \mathbf{0}$$

and determine the basis functions $\mathbf{P}$ and $\mathbf{Q}$. Taking into account the standard assumption, compare section 2, that $\mathbf{L}(\mathbf{i}_{\mathrm{L}})$ and $\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})$ are positive definite, we find[16]

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{\mathrm{C}} & \\ & \mathbf{0} \\ & \mathbf{I} \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathrm{C}} & \\ & \mathbf{I} \\ & \mathbf{0} \end{bmatrix},$$

where $\operatorname{im}\mathbf{Q}_{\mathrm{C}} = \ker\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}$, $\mathbf{0}$ and $\mathbf{I}$ are zero and identity matrices of appropriate dimension and the columns of $\mathbf{Q}_{\mathrm{C}}$ and $\mathbf{P}_{\mathrm{C}}$ together form a basis of $\mathbb{R}^{n_{\varphi}}$ with n_{φ} the dimension of $\boldsymbol{\varphi}$. The description of $\mathbf{Q}_{\mathrm{C}}$ results from the fact that (with $\mathbf{y} := \mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\mathbf{x}$)

$$\mathbf{x}^{\mathsf{T}}\mathbf{A}_{\mathrm{C}}\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\mathbf{x} = \mathbf{y}^{\mathsf{T}}\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})\mathbf{y} > 0, \quad \forall\mathbf{y} \in \mathbb{R}^{n_{\varphi}} \setminus \{\mathbf{0}\},$$

since $\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})$ is positive definite and hence the kernel of $\mathbf{A}_{\mathrm{C}}\mathbf{C}(\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\varphi})\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}$ is determined by the kernel of $\mathbf{A}_{\mathrm{C}}^{\mathsf{T}}$. As $\mathbf{A}_{\mathrm{C}}$ is an incidence (constant) matrix, we find that $\mathbf{Q}$ and $\mathbf{P}$ are also constant.

First example circuit

We now demonstrate the dissection concept by applying it to the example circuit given in Figure 1. The circuit contains a voltage source $v_{\mathrm{s}}(t)$, a linear resistor with conductance G, a linear capacitor with capacitance C, a linear inductor with inductance L, as well as a diode D that is modeled by a nonlinear resistance $g_{\mathrm{D}}(\varphi_3)$. Comparing the conditions of the topological index result from section 2 with the example circuit shows that the circuit has index $\nu = 1$, thus we only have to perform the first step of the dissection concept.
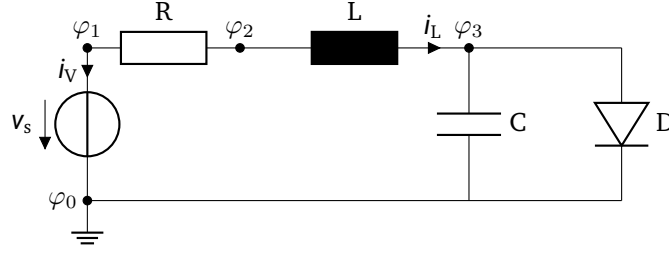
Figure 1 First example circuit: simple diode oscillator.

We begin by writing out the system derived from applying MNA to the example

$$\begin{bmatrix} 0 \\ & 0 \\ & & C \\ & & & L \\ & & & & 0 \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ i_L \\ i_V \end{bmatrix} + \begin{bmatrix} G & -G & & & 1 \\ -G & G & & & 1 \\ & & g_D(\varphi_3) & -1 \\ & & -1 & 1 \\ -1 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ i_L \\ i_V \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ v_s(t) \end{bmatrix} = \mathbf{0}.$$

For the first basis functions $\mathbf{Q}$ and $\mathbf{P}$ we find

$$\mathbf{Q} = \begin{bmatrix} 1 \\ & 1 \\ & & 0 \\ & & & 0 \\ & & & & 1 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

and as the matrix equivalent to $\mathbf{M}$ in the context of (5) is symmetric, we have $\mathbf{W} = \mathbf{Q}$ and $\mathbf{V} = \mathbf{P}$ for the remaining two basis functions. Using these, we find the new unknowns $\tilde{\mathbf{x}} = [\varphi_3 \ i_L]^\mathsf{T}$ and $\bar{\mathbf{x}} = [\varphi_1 \ \varphi_2 \ i_V]^\mathsf{T}$, which allow us to obtain the systems corresponding to (8a) and (8b)

$$\begin{bmatrix} C \\ & L \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} g_D(\varphi_3) & -1 \\ & 1 \end{bmatrix} \begin{bmatrix} \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & & \\ & -1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ i_V \end{bmatrix} = \mathbf{0} \tag{14a}$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} G & -G & 1 \\ -G & G & \\ -1 & & \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ i_V \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ v_s \end{bmatrix} = \mathbf{0}. \tag{14b}$$

Solving (14b) for $\bar{\mathbf{x}}$ and subsequently inserting the solution into (14a) then gives an ODE and a purely algebraic system as promised

$$\begin{bmatrix} C \\ & L \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} g_D(\varphi_3) & -1 \\ 1 & R \end{bmatrix} \begin{bmatrix} \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ -v_s \end{bmatrix} = \mathbf{0} \quad \text{and} \quad \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ i_V \end{bmatrix} = \begin{bmatrix} v_s \\ v_s - R i_L \\ -i_L \end{bmatrix}, \tag{15}$$

where $R = 1/G$ is the inverse of the conductance.

Second example circuit

We derive a second example from the first by substituting a current source $i_s(t)$ for the voltage source, compare Figure 2. Looking at the index criteria from section 2, we observe that this circuit has index $\nu = 2$, as there now is a cutset consisting of the inductor and current source. Therefore, we need to execute two steps of the dissection concept in order to split the equations into purely differential and algebraic parts respectively.
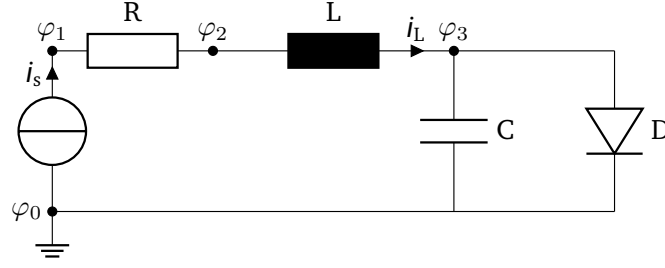
Figure 2 Second example circuit: simple diode oscillator with current instead of voltage source.

The corresponding MNA system is given by

$$\begin{bmatrix} 0 & & & \\ & 0 & & \\ & & C & \\ & & & L \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} G & -G & & \\ -G & G & & 1 \\ & & g_D(\varphi_3) & -1 \\ & -1 & 1 & \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ i_L \end{bmatrix} + \begin{bmatrix} -i_s(t) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0},$$

where we also note that the dimension is smaller compared to the previous example. The first two basis functions $\mathbf{Q}$ and $\mathbf{P}$ are

$$\mathbf{Q} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

where it holds again that $\mathbf{W} = \mathbf{Q}$ and $\mathbf{V} = \mathbf{P}$ due to symmetry. We now list the remaining basis functions, omitting the intermediate steps for brevity,

$$\bar{\mathbf{Q}} = \bar{\mathbf{W}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \bar{\mathbf{P}} = \bar{\mathbf{V}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{Q}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{P}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{W}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{V}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and finally obtain an ODE in one variable only and a second purely algebraic system that recovers the remaining unknowns

$$C\frac{\mathrm{d}}{\mathrm{d}t}\varphi_3 + g_D(\varphi_3)\varphi_3 - i_s = 0 \quad \text{and} \quad \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ i_L \end{bmatrix} = \begin{bmatrix} \varphi_2 + Ri_s \\ \varphi_3 + L\frac{\mathrm{d}}{\mathrm{d}t}i_s \\ i_s \end{bmatrix}. \tag{16}$$

A key observation at this point is that all basis functions of the second step are also constant, as was the case for the first step. Although we did not continue the abstract decoupling process for MNA in subsection 3.3, this is indeed a general phenomenon. The decoupling can be organized based on topological conditions only [16], which agrees with the intuition given by the index result from section 2. We remark that a similar result for circuits including controlled sources exists, however only for the first stage of the decoupling, as it is framed in the context of semi-explicit methods for which one only requires a DAE in semi-explicit form [16]. An extension of this result to circuits of higher index is within the scope of further research.

4 | INDEX-AWARE LEARNING

In the following, we outline how to use the dissection concept in the context of machine learning. The workflow is illustrated in Figure 3 and directly follows the structure of the dissection concept. We give a general description in a first step, followed by examples using the two circuits from section 3.

1. Our approach begins by decoupling a given DAE into an ODE and a purely algebraic equation (AE) following the main steps of the dissection concept.

2. Afterwards, only the differential degrees of freedom (DOFs) of the ODE are learned. For a DAE of index ν, these would be the entries of $\tilde{\mathbf{x}}_\nu$ using the notation of section 3.
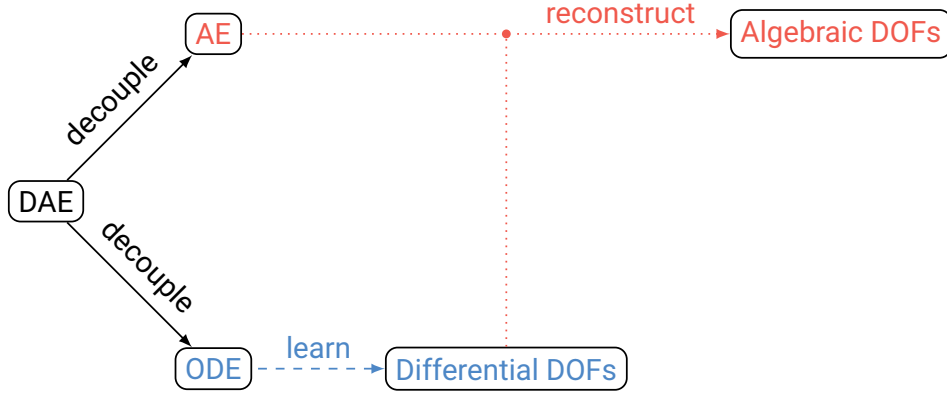
Figure 3 Schematic workflow of index-aware learning.

  3. The remaining algebraic DOFs $\bar{\mathbf{x}}_v$ may then be reconstructed using the algebraic equation.

We remark that the identification of the differential DOFs and thus the advantages of the second point are in principle available for any Spice based simulator via the topological decoupling[16], whereas the reconstruction of the algebraic DOFs requires an additional implementation. However, it is possible to trade some of the accuracy of our approach for an easier implementation, see subsection 5.1.

   In terms of the example circuits from subsection 3.3 the workflow amounts to the following: for the first example we only consider φ_3 and i_{L} as variables that need to be learned and for the second example solely φ_3 is left. All remaining degrees of freedom may be recovered using (15) or (16) respectively. Thus the learning effort is already reduced quite significantly in these two examples; from five to two variables in the first and from four down to one variable in the second example. Another key benefit comes from the fact that the reconstructed algebraic variables exactly fulfill the inherent constraints of the DAE. This means that even though the learned solution variables (think of φ_3 for example) are only approximations, the reconstructions (e.g. φ_1 or φ_2) will still be consistent. This may be of great importance for systems where the physical interpretability of the solution depends on it satisfying the constraints.

   There is yet another, maybe less expected, benefit that might occur. Looking at the decoupled system from (16) we find that the resistance parameter R and the inductance parameter L only appear in the algebraic equation. In terms of our original goal of speeding up design optimization or uncertainty quantification, where a lot of solutions for varying parameter values are required, this means the following: instead of needing to solve the full system for a given combination of R and L, we can instead merely solve the algebraic equation to obtain the full solution. While the algebraic equation might be more complicated than the simple linear relations of (15) and (16) in general, solving it is almost certainly much faster than having to integrate the entire system in time. This becomes an even bigger advantage when knowledge about the solution is only required for specific points in time, since the algebraic equation may be solved pointwise.

   Lastly, we emphasize that the approach is independent of the particular machine learning technique that is used for predicting the differential DOFs. Thus methods developed especially for ODEs may be employed and exchanged depending on the problem at hand.

5 | NUMERICAL EXAMPLES

Before we present numerical results regarding the two example circuits introduced in subsection 3.3, we want to provide some background on our machine learning method of choice, Gaussian processes (GPs), and the particular learning strategy we employ.

Gaussian processes

The following brief introduction is based on the textbook of Rasmussen and Williams[17] and we refer to the book itself for more details. We consider the problem of approximating one component $x(t)$ of the DAE solution $\mathbf{x}(t)$, compare (5), based on

observations

$$O = \{(t_i, x_i) : 1 \le i \le N\}.$$

We will focus on the one-dimensional case for clarity of exposition, however we remark that the ideas extend to the case where the solution component $x(t, \mathbf{p})$ also depends on the parameters and thus more than one variable, compare also the textbook [17]. A GP suitable for this problem is defined by a (prior) mean function $m : \mathbb{R} \to \mathbb{R}$ and covariance function $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The learning problem is then tackled using Bayesian inference, such that one aims to obtain the posterior distribution $\hat{x}(t)$, given the observations O and a point t, where x is to be predicted. A particular feature of GPs is that this posterior process, under suitable assumptions, turns out to be another GP with posterior mean and covariance [18]

$$\hat{m}(t) = m(t) + \mathbf{w}(t)^\top (\mathbf{x} - \mathbf{m}) \tag{17a}$$

$$\hat{k}(t) = k(t, t) - \mathbf{w}(t)^\top \mathbf{k}(t), \tag{17b}$$

where $\mathbf{m} := [m(t_1), \dots, m(t_N)]^\top$ denotes the prior mean function evaluated at the observations, $\mathbf{k}(t) := [k(t, t_1), \dots, k(t, t_N)]^\top$ similarly denotes the pairwise evaluation of the covariance function using the prediction point t and the observations, and $\mathbf{x} := [x_1, \dots, x_N]^\top$ are the observed function values. The weights $\mathbf{w}(t)$ are given by the solution of

$$(\mathbf{K} + \sigma^2 \mathbf{I}) \mathbf{w}(t) = \mathbf{k}(t),$$

where $[\mathbf{K}]_{i,j} := k(t_i, t_j)$ and σ^2 models i.i.d. additive Gaussian noise on the observations. For a discussion on modeling choices where the assumptions leading to (17) are not fulfilled, see the review article by Swiler et al. [19].

The key model component influencing the learning process is the covariance (or kernel) function k, since it determines the approximation properties of the GP. It encodes prior knowledge about the function $x(t)$ that is to be learned, such as differentiability or characteristic length scales. We opt for a radial basis function kernel, given by

$$k(t_i, t_j, \sigma_k, \ell) = \sigma_k^2 \exp\left(-\left(\frac{t_i - t_j}{\ell}\right)^2\right),$$

where σ_k and the length scale ℓ are hyperparameters, which allow for better approximation capabilities of the GP. The kernel is selected to match the differentiability of the solution.

In practice, the mean function m is often taken to be zero as the data is assumed standardized and the hyperparameters are then determined by minimizing the negative log likelihood [18]

$$-\log\big(p(\mathbf{x}|\sigma, \sigma_k, \ell)\big) = \frac{1}{2}\big(\mathbf{x}^\top \mathbf{K}(\sigma, \sigma_k, \ell)^{-1}\mathbf{x} + \log\big(\det \mathbf{K}(\sigma, \sigma_k, \ell)\big) + N\log(2\pi)\big),$$

where $[\mathbf{K}(\sigma, \sigma_k, \ell)]_{i,j} := k(t_i, t_j, \sigma_k, \ell) + \sigma^2$.

Learning strategy

The learning strategy aims to exploit one of the key features of GPs: that they provide both a mean prediction $\hat{m}$ and an associated variance estimate $\hat{k}$, as detailed in (17). We use these properties in conjunction with a greedy sample selection strategy, that starts out with a small number of training data points and adds additional samples based on the variance estimate of the GP. This idea is not new, see again the textbook of Rasmussen and Williams [17] for more references and details, however we still want to outline our particular approach to make the results better interpretable and reproducible.

Our implementation proceeds as follows:

1. We select a grid of time points $T = \{t_i : 1 \le i \le N_T\}$ and parameter values $P = \{\mathbf{p}_i : 1 \le i \le N_P\}$, where each $\mathbf{p}_i$ represents a specific combination of parameter values, for which we want to model the solution of the DAE using a GP.

2. We select a subset $D \subset T \times P$ and use the corresponding solutions for the initial training of a GP.

3. As a termination criterion, we compute the posterior mean $\hat{m}(t, \mathbf{p})$ using (17a) in all grid points $(t, \mathbf{p}) \in T \times P$ and check whether the relative prediction error

$$e := \frac{\|\hat{\mathbf{m}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2},$$

is below a desired tolerance, and if not, continue with 4. Here, $\hat{\mathbf{m}} := [\hat{m}(t_1, \mathbf{p}_1), \dots, \hat{m}(t_{N_T}, \mathbf{p}_{N_P})]$ collects the mean predictions for all $(t, \mathbf{p}) \in T \times P$ and $\mathbf{x}$ contains the corresponding simulated solution values.
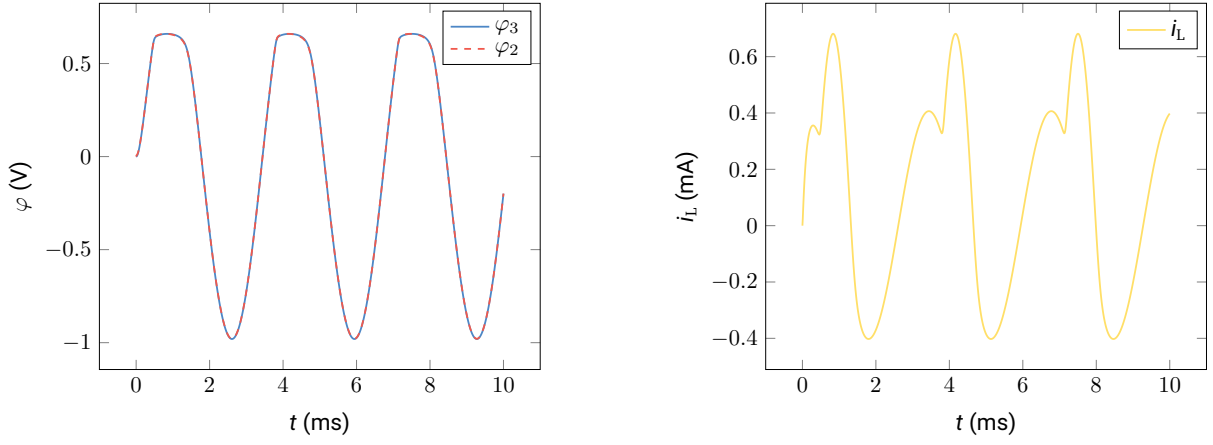
Figure 4 Solution of the first example circuit for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$.

4. We compute the variance prediction $\hat{k}(t, \mathbf{p})$ using (17b) for all $(t, \mathbf{p}) \in (T \times P) \setminus D$ and add the point of maximum variance to the training data set D.

5. Finally, we retrain the model and continue with 3.

We note that several improvements may be made to this strategy, such as properly maximizing the variance estimate in 4, instead of sampling on a discrete grid. The implementation is based on the STK toolbox[20].

First example circuit

We again consider the example of Figure 1, where we choose $R = 500\,\Omega$, $v_\text{s}(t) = \sin(600\pi t)\,\text{V}$ and the diode is modeled by $g_\text{D}(\varphi_3) = 10^{-14}\left(e^{(\varphi_3/26\,\text{mV})} - 1\right)\,\text{S}$. The remaining parameter values are chosen according to the greedy sample selection strategy with $1\,\text{mH} \leq L \leq 3\,\text{mH}$ and $100\,\text{nF} \leq C \leq 300\,\text{nF}$, i.e. $\mathbf{p} = [L\ C]$ in the context of section 1. The starting points for the strategy are given by all combinations of the boundary values for L and C together with $t = 0\,\text{ms}$ and $t = 10\,\text{ms}$ once for each combination. Recalling (15), we see that φ_3 and i_L have to be learned as the differential DOFs. In addition to these two, we also consider the algebraic variable φ_2 and we will compare the accuracy of learning φ_2 directly to recovering it from the two differential variables. Figure 4 shows the solution for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$, a combination which is not part of the training data.
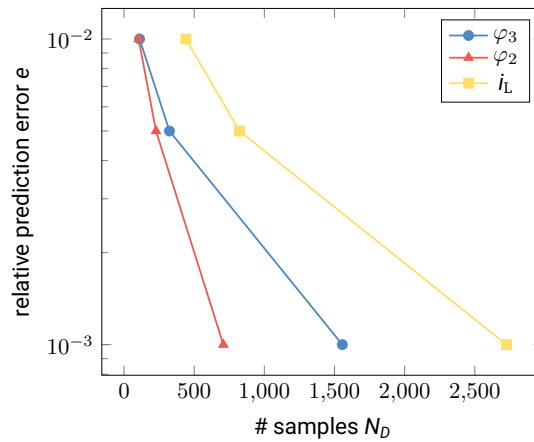

Figure 5 Convergence of the relative prediction error e, for the variables shown in Figure 4. The final accuracies correspond to approximately 180 different combinations of L and C for φ_3, 39 combinations for φ_2 and 37 for i_L.
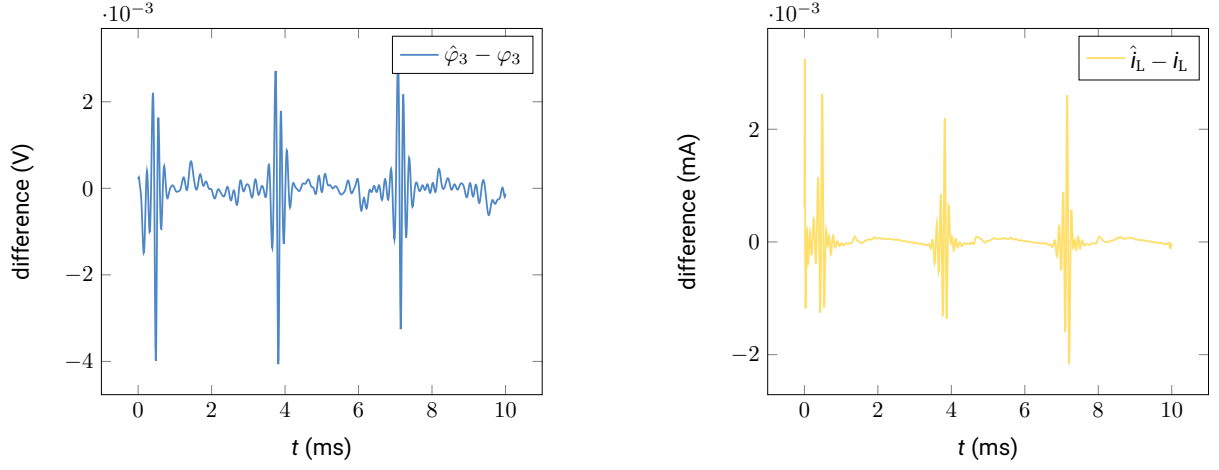
Figure 6 Differences between the mean predictions ($\hat{\varphi}_3$ and $\hat{i}_L$) and the corresponding simulations for the differential variables when considering $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$. The predictions correspond to the smallest relative errors from Figure 5.

In Figure 5, we see the convergence of the relative prediction errors with respect to the number of samples used by the greedy selection strategy. We observe that, depending on the qualitative complexity of the dynamics, the solution variables take different numbers of samples to reach the same prediction accuracy. It should be noted however, that the total number of samples N_D also includes the number of time points that are sampled, and not only the number of distinct parameter combinations (simulations). The latter are listed in the caption of Figure 5 and turn out to be considerably smaller. One may also note that we only execute the learning strategy up to a relatively large tolerance of 10^{-3}. This is due to the fact that both the optimization of the hyperparameters, as well as the computation of the posterior mean and variance, scale badly for conventional GPs, leading to large computation times. Remedies for this exist, see e.g. the book by Rasmussen and Williams[17], however this issue lies outside the scope of this article as our approach may be combined with any machine learning method of choice.

The differences between the mean predictions and simulations for the differential DOFs, when using the predictions belonging to the smallest relative errors from Figure 5, are shown in Figure 6. The results again correspond to $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$, and we observe that the differences are in line with the relative prediction errors of Figure 5. The differences between the mean prediction $\hat{\varphi}_2$, reconstruction $\bar{\varphi}_2$ and the simulation of the algebraic variable φ_2 are highlighted on the left of Figure 7. The
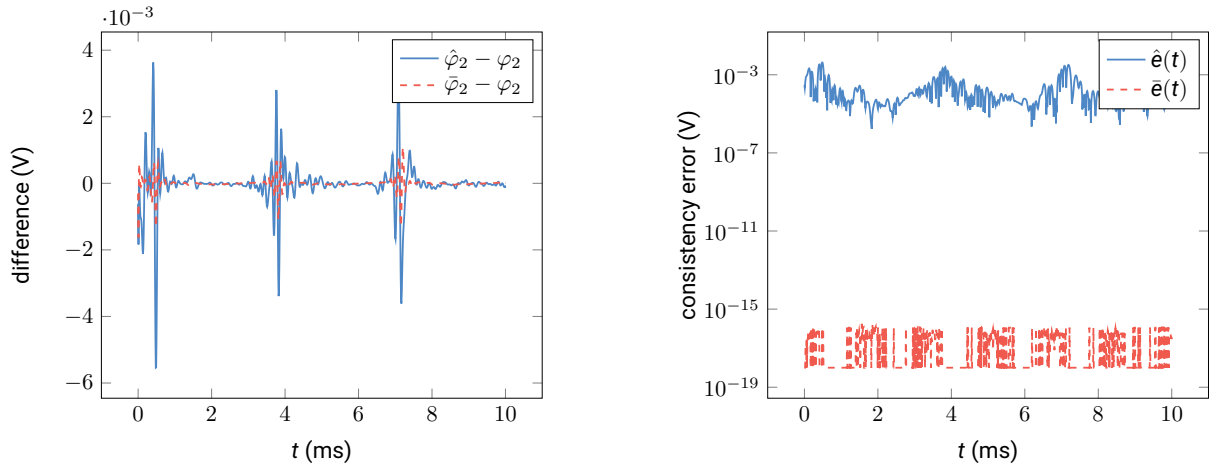
Figure 7 Differences between the mean prediction $\hat{\varphi}_2$, reconstruction $\bar{\varphi}_2$ and the corresponding simulation of the algebraic variable φ_2 (left) and the respective consistency errors (right). The predictions correspond to $L = 1.7\,\text{mH}$, $C = 220\,\text{nF}$ and the smallest relative error from Figure 5.

results again correspond to the same parameter values $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$. We observe that there is not much difference between the mean prediction and reconstruction, however the reconstruction does appear to have a slight advantage in terms of accuracy. Here, one should note that although the reconstruction is exact up to the accuracy of solving the algebraic equation (15), it still contains the error from learning the differential variables, hence the overall difference between $\bar{\varphi}_2$ and φ_2. To better quantify the difference between the learned and reconstructed solutions, we therefore introduce consistency errors $\hat{e}(t)$ and $\bar{e}(t)$ based on (15)

$$\hat{e}(t) := \left\| \begin{matrix} \hat{\varphi}_1 - v_\text{s} \\ \hat{\varphi}_2 + R\hat{i}_\text{L} - v_\text{s} \\ \hat{i}_\text{V} + i_\text{L} \end{matrix} \right\|_2, \qquad \bar{e}(t) := \left\| \begin{matrix} \bar{\varphi}_1 - v_\text{s} \\ \bar{\varphi}_2 + R\hat{i}_\text{L} - v_\text{s} \\ \bar{i}_\text{V} + i_\text{L} \end{matrix} \right\|_2,$$

where $\hat{}$ indicates learned quantities and $\bar{}$ refers to reconstructed quantities. For a consistent solution obeying all algebraic constraints, both errors are identically zero. The right of Figure 7 shows these consistency errors when using the same predictions and reconstructions as on the left. We observe a very clear improvement of the reconstructions (consistency error on the order of machine precision) over the directly learned predictions (consistency error as large as 10^{-3}). While the central benefit is of course this adherence to the constraints, only needing to learn two of the five solution variables also represents a significant reduction of the learning effort in this case.

Second example circuit

For the second example from Figure 2, we select $i_\text{s}(t) = 10^{-4}\sin(400\pi t)\,\text{V}$, while all other parameters and the learning strategy remain the same. Recalling (16), we find that φ_3 is the only differential variable that needs to be learned to reconstruct the remaining algebraic DOFs. We again focus on φ_2 as an algebraic variable to obtain a comparative example.

Considering similar numerical studies as for the first example, the left panel of Figure 8 shows the solutions of φ_3 and φ_2 for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$, which are again not part of the training data. The right plot shows the convergence of the relative prediction errors with respect to the total number of samples N_D. We again note that the number of distinct parameter combinations is significantly smaller, as listed in the caption. We also see that the relative errors behave similarly for both variables in this case, which is to be expected given the similar outlook of their dynamics in the left plot.

At this point we also return to the discussion from section 4 about parameters only appearing in the algebraic equation. During the learning process, the greedy strategy requested 21 unique values for C, all of which required full simulations according to (16). The 7 unique values for L, that were requested for learning φ_2, did not require full simulations however, but rather could be reconstructed from (16). We again emphasize that time is also included as a parameter within the sample selection strategy, such that the reconstruction not only constitutes a reduction from solving a nonlinear DAE in four variables to merely evaluating a scalar equation, but that this scalar equation furthermore only needs to be evaluated at the particular points in time that are requested by the strategy, rather than at all time points of the solution.
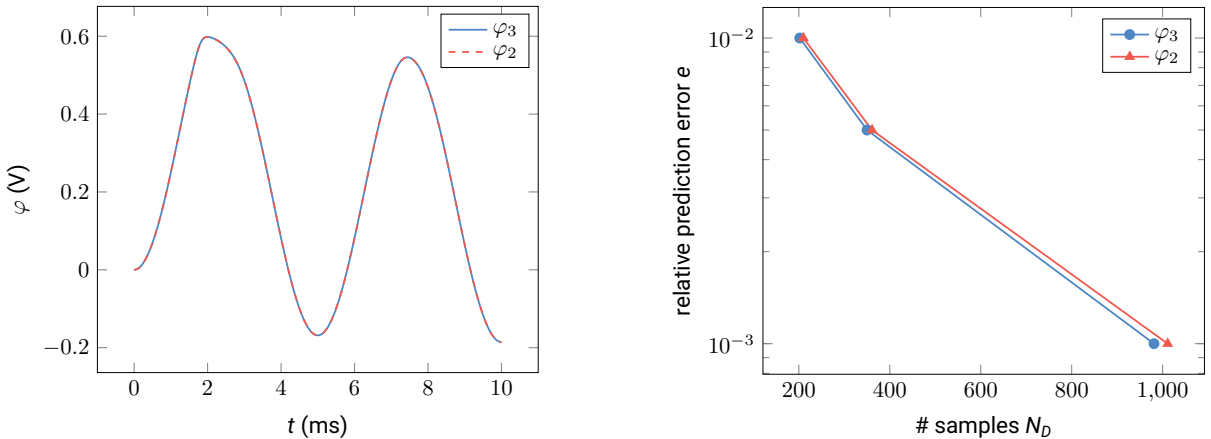
Figure 8 Solution of the second example circuit for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$ (left) and convergence of the relative prediction error for the same variables (right). The final accuracies correspond to 31 distinct combinations of L and C for φ_3 and 49 for φ_2.
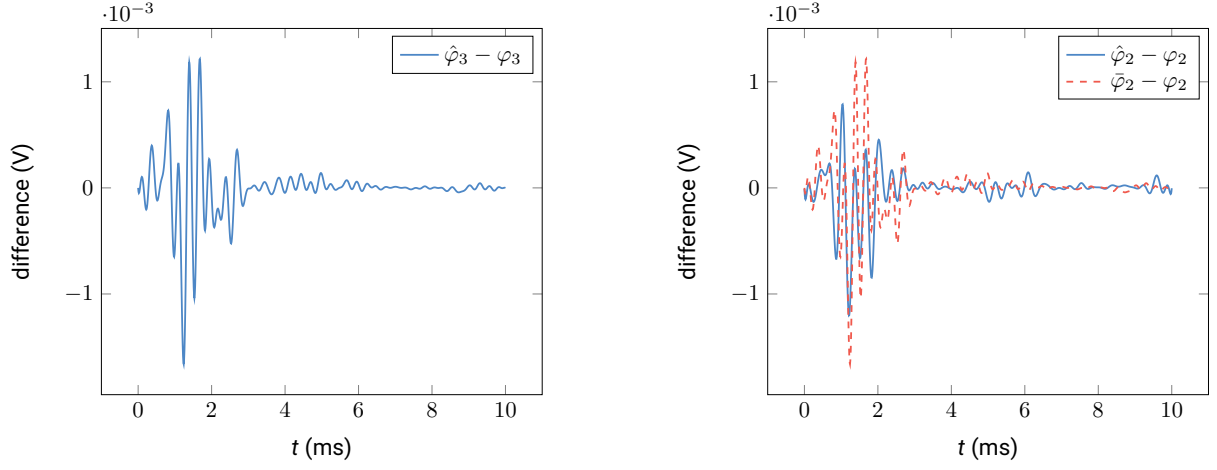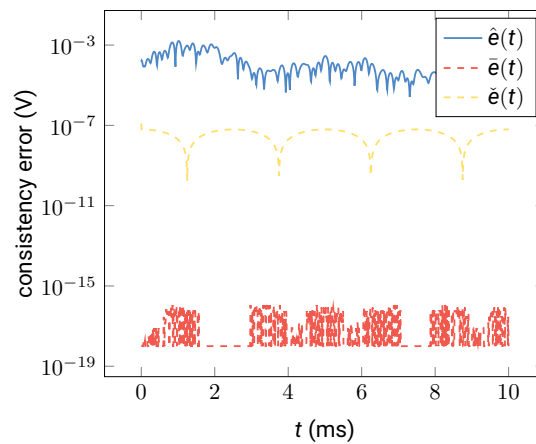
Figure 9 Differences between the mean prediction and simulation for φ_3 (left) and between the mean prediction, reconstruction and simulation for φ_2 (right). The predictions are again made for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$ and correspond to the smallest relative errors from Figure 8.
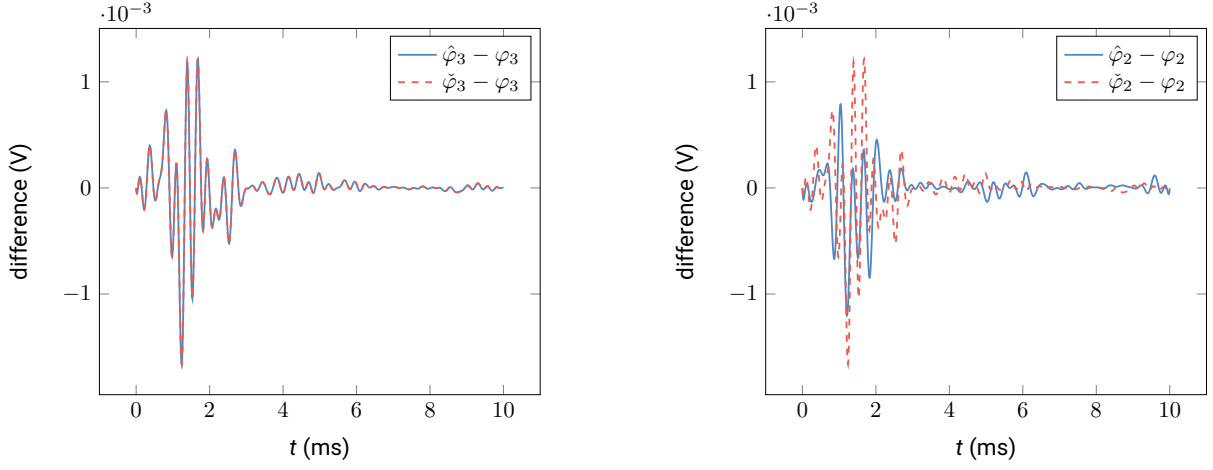
The difference between the mean prediction and simulation of φ_3, for $L = 1.7\,\text{mH}$ and $C = 220\,\text{nF}$, is shown on the left of Figure 9, while the right shows the differences between the mean prediction, reconstruction and simulation of φ_2. The predictions again correspond to the smallest relative errors from Figure 8 and the differences are of the same order. In this case, the reconstruction performs similar or even slightly worse compared to the mean prediction when only looking at the difference. However it still constitutes a reduction in learning effort, from four solution variables down to only one, and most importantly it adheres to the algebraic constraints of the DAE. To further illustrate this point, we again take a look at the consistency errors, now redefined based on (16)

$$\hat{e}(t) := \left\| \begin{matrix} \hat{\varphi}_1 - \hat{\varphi}_2 + Ri_\text{s} \\ \hat{\varphi}_2 - \hat{\varphi}_3 + L\frac{\text{d}}{\text{d}t}i_\text{s} \\ \hat{i}_\text{L} - i_\text{s} \end{matrix} \right\|_2 , \quad \bar{e}(t) := \left\| \begin{matrix} \bar{\varphi}_1 - \bar{\varphi}_2 + Ri_\text{s} \\ \bar{\varphi}_2 - \hat{\varphi}_3 + L\frac{\text{d}}{\text{d}t}i_\text{s} \\ \bar{i}_\text{L} - i_\text{s} \end{matrix} \right\|_2 .$$

In our example, the algebraic equation from (16) gives an explicit description of the algebraic variables once more, such that the reconstruction is accurate up to machine precision, compare $\bar{e}(t)$ in Figure 10. When learning all solution variables individually however, we observe a much larger maximum value of around 10^{-3} for the consistency error $\hat{e}(t)$ across all the predicted points

Figure 10 Consistency errors $\hat{e}(t)$ and $\bar{e}(t)$ corresponding to the results from Figure 9. The additional term $\check{e}(t)$ refers to the consistency error of the alternative approach, when using the predictions and Euler corrections outlined in Figure 11.

Figure 11 Differences between the mean predictions, Euler corrections and simulations for φ_3 (left) and φ_2 (right). The predictions are made for $L = 1.7\,\mathrm{mH}$ and $C = 220\,\mathrm{nF}$ and correspond to the smallest relative errors from Figure 8.

in time. In general, the algebraic equation may be nonlinear such that the reconstruction still leads to a consistency error $\bar{e}(t)$ greater than machine precision, depending on the accuracy of the nonlinear system solver.

5.1 | Alternative approach

In a last step, we want to provide an alternative approach to reducing the consistency error, while avoiding an extra implementation to handle the algebraic constraints. The idea is based on the insight that two successive implicit Euler steps suffice to bring the algebraic variables into a consistent state with the differential ones, when starting with inconsistent initial values [21]. In our context, one can then simply learn all solution variables independently, and when making a prediction, select a time point slightly before the intended prediction point, such that two small implicit Euler steps may be performed as a correction to obtain a consistent solution.

This alternative approach has the advantage that it can be directly applied using simulations from any SPICE-like software, and using any machine learning method for the predictions, without worrying about the reconstruction of the algebraic variables. However it may not achieve the same level of accuracy as the approach outlined in section 4 in practice, depending on the Euler step size and the accuracy of the nonlinear system solver [21]. Furthermore, it requires predictions for all solution variables, which represents unnecessary computational cost, and there is no chance for parameters to shift completely to the algebraic equations, compare section 4.

Figure 11 shows the differences between the mean predictions ($\hat{\varphi}_3, \hat{\varphi}_2$), Euler corrections ($\check{\varphi}_3, \check{\varphi}_2$) and the simulations for the second example, using the same parameters as in Figure 9. We see that the two Euler steps have almost no effect on the differential variable φ_3, as expected. For the algebraic variable φ_2 however, the mean predictions $\hat{\varphi}_2$ and the corrected predictions after two Euler steps $\check{\varphi}_2$, differ significantly, with the latter resembling the reconstruction from Figure 9. Also defining a consistency error $\check{e}(t)$ based on (16) in this case

$$\check{e}(t) := \left\| \begin{matrix} \check{\varphi}_1 - \check{\varphi}_2 + Ri_\mathrm{s} \\ \check{\varphi}_2 - \check{\varphi}_3 + L\frac{\mathrm{d}}{\mathrm{d}t}i_\mathrm{s} \\ \check{i}_\mathrm{L} - i_\mathrm{s} \end{matrix} \right\|_2$$

and comparing it to the other approaches, see Figure 10, shows that this alternative approach represents a compromise in terms of accuracy in the algebraic constraints. The maximum value of the consistency error $\check{e}(t)$ of the corrected predictions comes out around 10^{-7}, about four orders of magnitude smaller compared to before the correction. While it is theoretically possible to also force the error down to machine precision in this case, other factors become prohibitive as mentioned earlier.

6 | CONCLUSIONS AND FUTURE RESEARCH

This article introduced a new method for learning the time and parameter dependent behavior of electrical circuits. The method assumes the circuit to be modeled using MNA and then exploits the structure of MNA to improve the learning process by splitting it into a learning and a reconstruction stage, compare Figure 3. It achieves this by decoupling the underlying DAE into an ODE and a purely algebraic equation. Benefits of the method include a reduction in the number of variables that are to be learned during the learning stage and the exact adherence of the learned solution to the inherent constraints of the circuit model after the reconstruction stage. Two numerical examples illustrated both benefits. The examples also showed that the exact recovery of the constraints might improve the accuracy of the learned solutions. Furthermore, additional computational savings may be possible as some of the parameters of interest only appear in the reconstruction stage, which entirely avoids the need for training data with varying values of these parameters. We emphasize that the approach is independent of the machine learning method used during the learning stage, such that the learning method may be chosen according to the problem at hand. Lastly, we showcased an alternative approach that avoids the need for the decoupling and the associated implementation, while still improving the adherence of the learned solution to the algebraic constraints of the DAE.

Multiple extensions are possible within this index-aware learning framework. A natural first step could be to make use of the topological decomposition that was hinted at in section 3. This would pave the way for the inclusion of controlled sources within the workflow, and thus the industrial use of the approach, with the idea being the extension of the topological decomposition to also allow for controlled sources. Focusing in on the idea of adhering to physically meaningful constraints, one could work on extending the concept to a charge conserving variant of MNA, to potentially guarantee charge conservation even for the learned solutions. Yet another direction may be the application of index-aware learning to DAEs arising elsewhere, e.g. in modified loop analysis. (The dissection concept applies to a more general class of DAEs, as section 3 showed.) Finally, separate but related work may focus on improving the learning stage by developing new methods that are especially suited for learning ODEs.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Schilders WHA, Vorst V. dHA, Rommes J. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer . 2008

2. Xiu D. *Numerical Methods for Stochastic Computations*. Princeton University Press . 2010

3. Beneventano P, Cheridito P, Jentzen A, von Wurstemberger P. High-dimensional approximation spaces of artificial neural networks and applications to partial differential equations. 2020.

4. Zaabab AH, Zhang QJ, Nakhla M. A neural network modeling approach to circuit optimization and statistical design. *IEEE Transactions on Microwave Theory and Techniques* 1995; 43(6): 1349-1358. doi: 10.1109/22.390193

5. Fang Y, Yagoub MCE, Wang F, Zhang QJ. A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks. *IEEE Transactions on Microwave Theory and Techniques* 2000; 48(12): 2335-2344. doi: 10.1109/22.898982

6. Wang M, Lv W, Yang F, et al. Efficient Yield Optimization for Analog and SRAM Circuits via Gaussian Process Regression and Adaptive Yield Estimation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2018; 37(10): 1929-1942. doi: 10.1109/TCAD.2017.2778061

7. Sanabria-Borbón AC, Soto-Aguilar S, Estrada-López JJ, Allaire D, Sánchez-Sinencio E. Gaussian-Process-Based Surrogate for Optimization-Aided and Process-Variations-Aware Analog Circuit Design. *Electronics* 2020; 9(4). doi: 10.3390/electronics9040685

8. Ho CW, Ruehli A, Brennan P. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems* 1975; 22(6): 504-509. doi: 10.1109/TCS.1975.1084079

9. Analog Devices . LTspice. 2022.

10. Sandia National Laboratories . Xyce. 2022.

11. Cadence Design Systems . PSpice. 2022.

12. Denk G, Rentrop P. Mathematical models in electric circuit simulation and their numerical treatment. In: Strehmel K. , ed. *Numerical Treatment of Differential Equations. Proceedings of the fifth seminar "NUMDIFF-5"*Teubner-Texte zur Mathematik. Teubner-Verlag. 1991 (pp. 305-316).

13. Mehrmann V. *Index Concepts for Differential-Algebraic Equations*. Springer International Publishing . 2015

14. Tischendorf C. Topological index calculation of differential-algebraic equations in circuit simulation. *Surveys on Mathematics for Industry* 1999; 8: 187-199.

15. Estévez Schwarz D, Tischendorf C. Structural analysis of electric circuits and consequences for MNA. *International Journal of Circuit Theory and Applications* 2000; 28(2): 131-162.

16. Jansen L. *A Dissection Concept for DAEs*. PhD thesis. 2014

17. Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. The MIT Press . 2006.

18. Basak S, Petit S, Bect J, Vazquez E. Numerical issues in maximum likelihood parameter estimation for Gaussian process interpolation. In: 7th International Conference on Machine Learning, Optimization and Data science (LOD 2021). 2021.

19. Swiler LP, Gulian M, Frankel AL, Safta C, Jakeman JD. A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. *Journal of Machine Learning for Modeling and Computing* 2020; 1(2): 119-156. doi: 10.1615/JMachLearnModelComput.2020035155

20. Bect J, Vazquez E. STK: a Small (Matlab/Octave) Toolbox for Kriging. 2022.

21. Garcia IC, Kulchytska-Ruchka I, Schöps S. Parareal for index two differential algebraic equations. *Numerical Algorithms* 2022; 91(1): 389-412. doi: 10.1007/s11075-022-01267-1

AUTHOR BIOGRAPHY

Idoia Cortes Garcia received her Bachelor in mathematics and Master in modelling for science and engineering from the Autonomous University of Barcelona. She obtained her Ph.D. in electrical engineering from the Technical University of Darmstadt in 2020. Since 2022 she is an assistant professor at the group of Dynamics and Control from the department of mechanical engineering at Eindhoven University of Technology. Her research interests include coupled multiphysical dynamical systems, differential algebraic equations, efficient time domain (co-)simulation methods and hybrid modelling approaches.
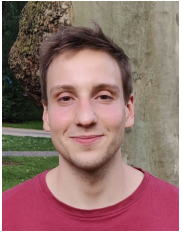
Peter Förster received his B.Sc. and M.Sc. degrees in electrical engineering from Technical University of Darmstadt. Currently, he is a doctoral researcher at the centre for analysis, scientific computing and applications at Eindhoven University of Technology and in the computational electromagnetics group at Technical University of Darmstadt. His research interests include circuit simulation, differential-algebraic equations, machine learning and digital twins.

Lennart Jansen received his M.Sc. degree in mathematics from Universität zu Köln and his Ph.D. degree in mathematics from Universität zu Berlin. Afterwards, he held a post-doc position at Heinrich-Heine-Universität in Düsseldorf before starting work in the industry. One of the main research topics he worked on in the industry was AI-based recommendations for design improvement in the automobile industry. He switched to the photovoltaic industry in 2021 and currently works there on optimal control for combined electric-water-gas networks.

Wil Schilders received the M.Sc. degree in mathematics from Radboud University Nijmegen, Nijmegen, The Netherlands, in 1978, and the Ph.D. degree from the Trinity College Dublin, Dublin, Ireland, in 1980. He has been working in the electronics industry with Philips Research Laboratories, Eindhoven, The Netherlands, since 1980, and NXP, since 2006, where he developed algorithms for simulating semiconductor devices, electronic circuits, organic light emitting diodes, and electromagnetic problems (TV tubes, interconnects, and magnetic resonance imaging). Since 1999, he has been a Part-Time Professor of numerical mathematics for industry with the Technical University of Eindhoven, Eindhoven. He is currently the Managing Director of the Platform for Mathematics in The Netherlands. He is also the President of the European Consortium for Mathematics in Industry.

Sebastian Schöps received the M.Sc. degree in business mathematics and the joint Ph.D. degree from Bergische Universität Wuppertal and Katholieke Universiteit Leuven in mathematics and physics. He was appointed as a Professor of Computational Electromagnetics at Technische Universität Darmstadt within the interdisciplinary center of computational engineering, in 2012. His current research interests include coupled multi-physical problems, bridging computer aided design and simulation, parallel algorithms for high performance computing, digital twins, uncertainty quantification, and machine learning.