Interpretable large-signal black-box models of power electronic converters: a multimodel approach

Antonin Colot, Simone Paoletti, Antonio Giannitrapani, Mevludin Glavic, and Bertrand Cornélusse

***Abstract*—In this paper, we propose a multimodel approach to derive large-signal black-box models of power converters that are suitable for system-level studies. We introduce an interpretable black-box large-signal model by taking advantage of the inherent interpretability of linear system models and the design of an interpretable weighting function. First, the non-linear model of a power converter is approximated by a set of linear submodels. Next, we consider a neural network-based weighting function trained to combine the linear submodels' responses. A post-analysis of the trained neural network is used to speed up the partitioning of the operating space by restricting the number of new experiments that have to be carried out. A single-machine infinite bus system is used to illustrate the rationale behind the proposed multimodel approach and to illustrate some of its inherent limitations. The overall methodology is illustrated using a voltage-regulated DC-DC boost converter. Finally, the approach is validated using a small system including a battery, a voltage-regulated DC-DC boost converter, and a DC motor.**

***Index Terms*—Power electronic converters, Large-signal black-box model, Multimodel approach, Interpretability**

I. Introduction

The increasing integration of power electronic converters (PECs) in modern power systems introduces fast and complex non-linear dynamics [1], contrasting with the traditionally slower dynamics driven by electromechanical phenomena. This paradigm shift requires new time-domain simulation tools for assessing the stability of power systems under various operating conditions [2], [3].

Modeling PECs usually requires detailed information about their internal parameters and electrical configuration [4]. However, modern power systems often incorporate *Off-The-Shelf* converters from different manufacturers [5], [6] with limited available data provided to the user. This calls for the development of black-box models [7], which relate inputs and outputs of systems without requiring any knowledge of their internal mechanism.

Generally, PECs are non-linear, time-varying systems [2]. Switching models of PECs are not relevant for system-level studies because they have to be solved with small time steps, which is a prohibiting factor for simulating large-scale systems. We focus on state-space averaged models of PEC that are time-invariant, and thus, can be solved using much larger time steps than switching models, while giving similar results for low-frequency dynamics [8]. Notice that averaged models of PEC remain non-linear, and non-linear system identification is much more challenging than linear system identification as far as the number of possible model structures, and experiment design, as well as the difficulty of parameter estimation are concerned [9]. On the other hand, linear models have some limitations when considering PECs, as they cannot adequately capture their large-signal behaviors. Although one can use PEC linear models to design new controllers [2], this is not appropriate for system-level studies since PECs' dynamics may depend heavily on the operating point.

This paper introduces a multimodel approach to approximate the behavior of state-space averaged models of PEC and derive large-signal black-box models for system-level design and analysis. The multimodel approach computes the global model output as a weighted combination of some linear submodel responses. The weighting function represents the linear submodel's validity within a specific operating range [10]. This approach is simple, mathematically tractable, and built upon the well-known theory of linear system identification.

Literature review. In [11], authors compare different modeling techniques for PECs and show the limited capabilities of linear models when representing the global behavior of PECs. Many different model structures belong to the multimodel approach. Among them, polytopic models exhibit some advantages for modeling power converters. In [6], [7], authors propose a polytopic model with a double sigmoid weighting function for power converters, while [12] emphasizes the performance of polytopic models for system-level modeling. In [13], authors compare polytopic models with a wavelet convolutional neural network trained to represent the PEC behavioral response. In [14], authors use a recurrent neural network to approximate the PEC's dynamics. Although fully neural network-based models lead to better results, they lack interpretability and are highly dependent on data quality [15]. Ensuring sufficient data quality and availability may be challenging for PEC applications, and the lack of physical insights behind those full neural network-based models may make the analysis of power systems challenging. Other modeling techniques exist, such as Hammerstein models [16]. Those methods combine static non-linear blocks with dynamical linear systems, and are very convenient when the non-linearity stems from saturation, sensors and actuators, etc. [6]. However, when the non-linearity comes from the switching nature of the converter, the range of applicability of these methods is limited since the dynamics of the system become dependent on the

The work of Antonin Colot was supported by the Research Fellow Fellowship of the F.R.S-FNRS.

(Corresponding author: Antonin Colot.)

Antonin Colot, Mevludin Glavic and Bertrand Cornélusse are with Department of Electrical, Engineering and Computer Science, University of Liège, Liège, Belgium (email: (antonin.colot, mevludin.glavic, bertrand.cornelusse)@uliege.be).

Simone Paoletti and Antonio Giannitrapani are with Department of Information Engineering and Mathematics of the University of Siena, Siena, Italy (email: (paoletti, giannitrapani)@dii.unisi.it).

operating point.

Contributions. For a given multimodel structure, one has to face three challenges: how to identify the linear submodels, how to partition the operating space and how to design the weighting function. We address the last two challenges and provide some guidelines for dealing with the first one. The paper contributions can be summarized as follows:

- An interpretable neural network-based weighting function is trained to combine the linear submodels' responses to best fit the response of the power electronic converter. The weighting function interpretability is ensured by its dependence only on the inputs of the system. This also ensures tractability for high dimensional state systems.
- The partitioning of the operating space is based on the post-analysis of the weighting function. This partitioning reduces the number of experiments that should be carried out and significantly speeds up the identification procedure compared to other partitioning methods, e.g. LOLIMOT [17], for complex systems with multiple inputs and outputs.
- This paper extends the work done in [18] by introducing a state-space model formulation and a state update technique to enhance the performance when the input signals change abruptly. Moreover, a system-level study is presented to validate the multimodel approach for a small system.

The remainder of the paper is organized as follows. A general framework for the identification problem is presented in Section 2 while Section 3 illustrates the rationale behind our approach using a single-machine infinite bus system. Section 4 presents the black-box model of the DC-DC boost converter. Section 5 validates the proposed modeling approach using a small system, while Section 6 offers some conclusions.

II. General framework

We consider a discrete-time non-linear dynamical system of the form:

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \tag{1a}$$

$$\mathbf{y}(k) = h(\mathbf{x}(k), \mathbf{u}(k)), \tag{1b}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$,$\mathbf{u} \in \mathbb{R}^{n_u}$, $\mathbf{y} \in \mathbb{R}^{n_y}$ are the system state, input and output, respectively, $f : \mathbb{R}^{n_x \times n_u} \mapsto \mathbb{R}^{n_x}$, $h : \mathbb{R}^{n_x \times n_u} \mapsto \mathbb{R}^{n_y}$ is the output function and $\mathbf{x}(k), \mathbf{u}(k)$ and $\mathbf{y}(k)$ are the state, input and output values at continuous time $k\tau$ with $\tau \in \mathbb{R}$ the time step considered. We want to approximate the dynamics of the non-linear system (1) using a multimodel approach. To this purpose, we combine the outputs of the linear submodels through a weighting function [10]. This allows us to have heterogeneous submodels compared to other multimodel approaches where, for instance, the state of the global system is a weighted sum of the states of each submodel. In that case, the different linear submodels have to share the same state space dimensionality.

Let us consider the set $\mathcal{N} := \{1, ..., N\}$ of indices of the different linear submodels. The linear submodel $i \in \mathcal{N}$, obtained around the operating point $(\mathbf{u}_i^e, \mathbf{y}_i^e)$ via linearization techniques, is described by the following state-space equation:

$$\tilde{\mathbf{x}}_i(k+1) = A_i\tilde{\mathbf{x}}_i(k) + B_i\tilde{\mathbf{u}}_i(k), \tag{2a}$$

$$\mathbf{y}_i(k) = C_i\tilde{\mathbf{x}}_i(k) + D_i\tilde{\mathbf{u}}_i(k) + H_i, \tag{2b}$$

with $\tilde{\mathbf{u}}_i := \mathbf{u} - \mathbf{u}_i^e \in \mathbb{R}^{n_u}$, $\tilde{\mathbf{x}}_i \in \mathbb{R}^{n_{x,i}}$,$\mathbf{y}_i \in \mathbb{R}^{n_y}$, and $A_i \in \mathbb{R}^{n_{x,i} \times n_{x,i}}$, $B_i \in \mathbb{R}^{n_{x,i} \times n_u}$, $C_i \in \mathbb{R}^{n_y \times n_{x,i}}$, $D_i \in \mathbb{R}^{n_y \times n_u}$, $H_i \in \mathbb{R}^{n_y}$.

Interpretability of black-box models is a vibrant research field and numerous approaches have been proposed to this purpose [19], [20]. We stick with the notion of interpretability focused on how the system outputs are affected by its inputs. Therefore, we capitalize on the fact that linear models are inherently interpretable (intuitive meaning of its parameters, they are additive, etc.) [19].

To further ensure interpretability of the proposed approach, in this paper we combine different linear submodels' outputs as follows:

$$\mathbf{y}_{\mathrm{PM}}(k) = \sum_{i \in \mathcal{N}} \omega_i(\mathbf{u}(k))\mathbf{y}_i(k), \tag{3}$$

with $\omega_i : \mathbb{R}^{n_u} \mapsto \mathbb{R}$ being the weighting function for the linear submodel i, such that $\omega_i(\mathbf{u}(k)) \in [0, 1]$ and $\sum_{i \in \mathcal{N}} \omega_i(\mathbf{u}(k)) = 1$ at every step k.

Notice that the weighting functions $\omega_i(\cdot)$ depend on $\mathbf{u}(k)$ only. This dependence on the inputs improves the global model interpretability. For one constant input $\mathbf{u}(k) = \mathbf{u}^*$, the linear submodel i^* mostly responsible for the global model behavior has the largest associated weight $\omega_{i^*}(\mathbf{u}^*)$. On the other hand, it leads to a suboptimal combination of submodel responses during "jumps" in the inputs. This drawback is discussed later.

The rationale behind the input-dependent weighting function used in (3) is fully detailed in the next section using the single-machine infinite bus system together with the illustration of some challenges related to the modeling of non-linear dynamical systems with proposed multimodel approach.

III. The Single-Machine Infinite Bus (SMIB) system

The SMIB system consists of one equivalent synchronous machine connected to an infinite bus through a line of reactance X. The dynamics of the equivalent synchronous machine is described with the swing equations. Although simple, this system is often used in power system practice through one-machine infinite bus transformation for angle stability problems [21].

Consider the SMIB system:

$$\delta(k+1) = \delta(k) + \tau\omega_{\mathrm{s}}\Delta\omega(k), \tag{4a}$$

$$\Delta\omega(k+1) = \Delta\omega(k) + \frac{\tau}{M}\left(P_{\mathrm{m}} - D\Delta\omega(k) - \frac{VV_\infty \sin(\delta(k))}{X}\right), \tag{4b}$$

where δ is the rotor angle of the single machine, $\Delta\omega$ the deviation of the angular speed from the nominal angular speed of the machine in per unit, P_{m} is the mechanical power in per unit, D the damping parameter, X the reactance of the line connecting the single machine to the infinite bus

system, and M the moment of inertia of the machine. For this system, let us consider the input $u = V_\infty$, the output $\mathbf{y}(k) = [\delta(k), \Delta\omega(k)]^\top$, the state vector $\mathbf{x}(k) = [\delta(k), \Delta\omega(k)]^\top$, and the parameters $\omega_\mathrm{s} = 2\pi 60$ [rads^{-1}], $P_\mathrm{m} = 0.5$ [pu], $D = 5$ [pu], $V = 1$ [pu], $X = 1.5$ [pu], $M = 8$ [s]. Finally, let us consider $\tau = 1e-2$ [s].

One equilibrium point $\mathbf{x}^e = [\delta^e, \Delta\omega^e]^\top$ of the system (4) is:

$$\delta^e = \arcsin\left(\frac{P_\mathrm{m} X}{V V_\infty}\right) \in [0, \pi/2] \tag{5a}$$

$$\Delta\omega^e = 0 \tag{5b}$$

A. Linear approximations

We assume that $V_\infty \in [0.85, 1.15]$, and we define $N = 3$ linear systems around the equilibrium points $x_i^e = [\delta_i^e, 0]^\top$ as follows:

$$A_i = \begin{bmatrix} 1 & \tau\omega_\mathrm{s} \\ -\tau\frac{V V_{\infty,i}\cos(\delta_i^e)}{MX} & -\tau\frac{D}{M}+1 \end{bmatrix}, B_i = \begin{bmatrix} 0 \\ -\tau\frac{V\sin(\delta_i^e)}{XM} \end{bmatrix}, \tag{6a}$$

$$C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, H_i = \begin{bmatrix} \delta_i^e \\ 0 \end{bmatrix}, \tag{6b}$$

where $V_{\infty,1} = 0.9, V_{\infty,2} = 1.0, V_{\infty,3} = 1.1$ and $\delta_i^e \ \forall i \in \mathcal{N}$ are obtained from equation (5a).

B. Weighting functions

In this example, we consider two types of weighting functions for the polytopic model (2)– (3).

a) Piecewise-Affine (PWA): Let us consider the case where only one linear submodel is active for a given input u. We define $d = 0.1$, that is the difference between two input linearization points, *i.e.*, $V_{\infty,i} - V_{\infty,i-1} \ \forall i \in \{2, 3\}$. In this simple example, notice that the input operating space $u \in [0.85, 1.15]$ is divided into three similar operating regions, *i.e.*, linear submodel LM0 should be valid for $u \in [0.85, 0.95]$, linear submodel LM1 for $u \in [0.95, 1.05]$ and linear submodel LM2 for $u \in [1.05, 1.15]$. Now, we write:

$$\omega_i(u) = \omega_i(V_\infty) = \begin{cases} 1 \text{ if } |V_\infty - V_{\infty,i}| \leq \frac{d}{2}, \\ 0 \text{ else.} \end{cases} \tag{7}$$

Notice that for $V_\infty = 0.95$ and $V_\infty = 1.05$, there are two weights equal to 1. In practice, we activate only one linear submodel to ensure that $\sum_{i\in\mathcal{N}} \omega_i(\mathbf{u}(k)) = 1$.

b) Double Sigmoid (DS): The second type of weighting function provides a smooth transition from one linear submodel to another when the input is slowly changing. We write:

$$\omega_i(V_\infty) = \frac{1}{1+\exp\left(-s_i^+\left(V_\infty - (V_{\infty,i} - d)\right)\right)} - \frac{1}{1+\exp\left(-s_i^-\left(V_\infty - (V_{\infty,i} + d)\right)\right)}, \tag{8}$$

where $s_i^+ \in \mathbb{R}$ and $s_i^- \in \mathbb{R}$ are the slopes of the double sigmoid, and should be sufficiently large such that $\omega_i(V_{\infty,i}) = \frac{1}{1+\exp(-s_i^+ d)} - \frac{1}{1+\exp(s_i^- d)} \approx 1$.

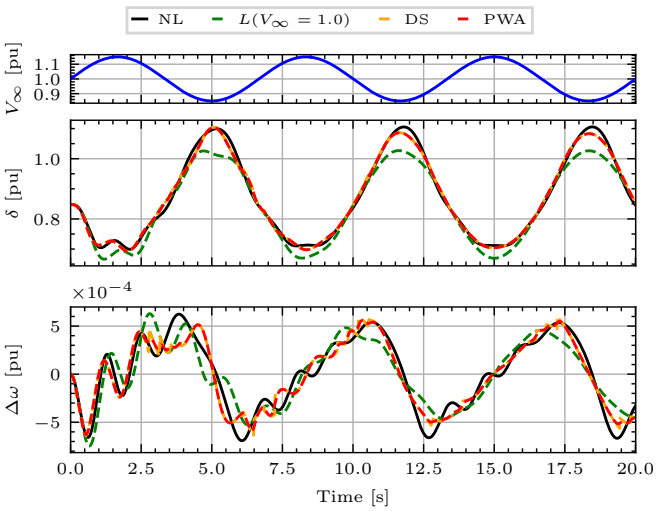


Fig. 1: Comparison for a slowly time-varying input between a linear model and polytopic models with two different weighting functions.

C. Comparison between polytopic and linear models

Let us first consider a slowly time-varying input $u(k) = 1 + 0.15\sin(0.3\pi k\tau)$, and compare the performance of a linear model identified for $V_\infty = 1$, denoted $L(V_\infty = 1.0)$, and that of two polytopic models with PWA and DS weighting functions. To evaluate the estimation accuracy, we use the coefficient of determination defined as:

$$R_j^2 = 1 - \frac{\|\mathbf{y}_j - \hat{\mathbf{y}}_j\|^2}{\|\mathbf{y}_j - \bar{y}_j\mathbf{1}\|^2}, \tag{9}$$

where $\hat{\mathbf{y}}_j$ is the vector including the samples $y_j(k)$ of the j^{th} output of the approximate model, while $\mathbf{y}_j$ corresponds to the j^{th} output of the true non-linear system and $\bar{y}_j$ the mean value of $\mathbf{y}_j$. For the SMIB system, recall that $\mathbf{y}(k) = [\mathbf{y}_1(k), \mathbf{y}_2(k)]^\top = [\delta(k), \Delta\omega(k)]^\top$. The $R^2 = [R_\delta^2, R_{\Delta\omega}^2]$ scores for the linear model, the PWA polytopic model and the DS polytopic model are [0.928,821], [0.993, 0.901], [0.993, 0.908], respectively. The simulation results are shown in Fig. 1 where NL is the output of the true nonlinear system.

We then consider a piecewise constant input signal. The R^2 scores for the linear model, the PWA polytopic model and the DS polytopic model are [0.673,0.047], [0.924,0.768], [0.924,0.768], respectively. Notice that PWA and DS polytopic models have the same R^2 scores. The simulation results are shown in Fig. 2.

One can notice that the polytopic models perform better than the linear model in both scenarios; for slowly time-varying inputs and step signals. However, when the input signal changes abruptly, the weights associated with each linear submodel also change quickly since the weighting function only depends on the input. As a consequence, when a previously non-active linear model becomes active, its state might be far away from the operating region where the linear submodel was identified. This creates "jumps" in the states and in the outputs of the global model, as shown in Fig. 3. In the next subsection, we introduce a state update technique to smooth out transitions from one linear submodel to another during "jumps" in the inputs.

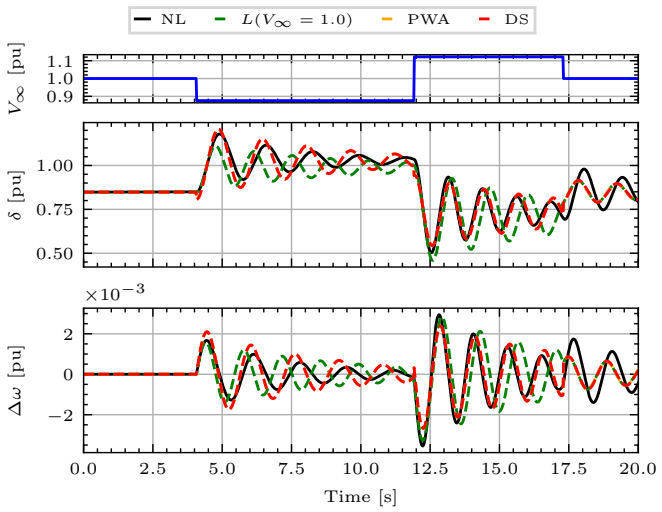


Fig. 2: Comparison for a piecewise constant input between a linear model and polytopic models with two different weighting functions.

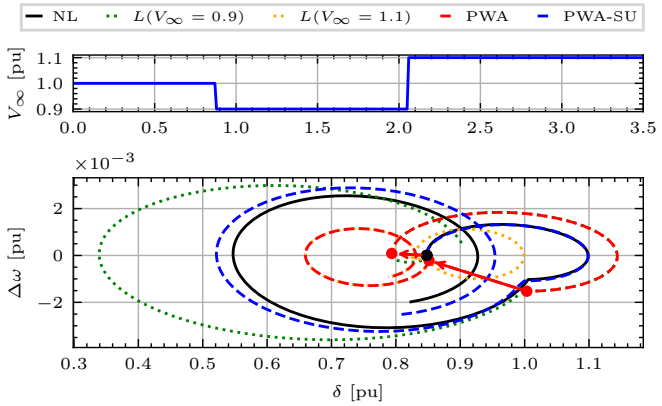


Fig. 3: Phase portrait comparison between different modeling techniques.

D. Modified state update

In our framework, the weighting function only depends on the inputs $\mathbf{u}$. This makes the partitioning in different operating regions, and the realization of experiments on the system, much easier. However, the polytopic model may perform poorly when the input changes quickly, as shown in Fig. 3. In this section, we investigate a state update technique to smooth out the response of the polytopic model for abrupt changes in the input. The idea is to find the state $\tilde{\mathbf{x}}_i^{\text{es}}(k)$ at every step k such that the distance between the output trajectory of the linear submodel and the one of the polytopic model is minimized. First, define $\mathbf{z}_{PM}(k-1) \in \mathbb{R}^{T\cdot n_y}, \mathbf{z}_i(k-1) \in \mathbb{R}^{T\cdot n_y}$ as the vectors that contain the T previous outputs of the polytopic model and of the linear submodel i, respectively:

$$\mathbf{z}_{PM}(k-1) = \begin{pmatrix} \mathbf{y}_{PM}(k-T) \\ \mathbf{y}_{PM}(k-T+1) \\ \vdots \\ \mathbf{y}_{PM}(k-1) \end{pmatrix},$$

$$\mathbf{z}_i(k-1) = \begin{pmatrix} \mathbf{y}_i(k-T) \\ \mathbf{y}_i(k-T+1) \\ \vdots \\ \mathbf{y}_i(k-1) \end{pmatrix}. \tag{10}$$

We consider the sequence $\{\tilde{\mathbf{x}}_i(k-T), \tilde{\mathbf{x}}_i(k-T+1), ..., \tilde{\mathbf{x}}_i(k-1)\}$ obtained via (2) that yields the output sequence $\mathbf{z}_i(k-1)$. We obtain the initial state $\tilde{\mathbf{x}}_i^{\text{es}}(k-T)$ by minimizing the difference between the output vector of the linear submodel i and the output vector of the polytopic model:

$$\tilde{\mathbf{x}}_i^{\text{es}}(k-T) = \arg\min_{\tilde{\mathbf{x}}_i(k-T)} \|\mathbf{z}_{PM}(k-1) - \mathbf{z}_i(k-1)\|^2, \tag{11}$$

where $\mathbf{z}_i(k-1)$ is a function of $\tilde{\mathbf{x}}_i(k-T)$. Using (2), we have

$$\mathbf{z}_i(k-1) = \begin{pmatrix} C_i \\ C_iA_i \\ \vdots \\ C_i(A_i)^{T-1} \end{pmatrix} \tilde{\mathbf{x}}_i(k-T) + \begin{pmatrix} D_i & & \\ C_iB_i & D_i & \\ \vdots & & \\ C_iA_i^{T-2}B_i & \dots & D_i \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{u}}_i(k-T) \\ \tilde{\mathbf{u}}_i(k-T+1) \\ \vdots \\ \tilde{\mathbf{u}}_i(k-1) \end{pmatrix} + \begin{pmatrix} H_i \\ H_i \\ \vdots \\ H_i \end{pmatrix} \tag{12}$$

that can be written as

$$\mathbf{z}_i(k-1) - R_i = \mathcal{O}_i\tilde{\mathbf{x}}_i(k-T), \tag{13}$$

with $\mathcal{O}_i \in \mathbb{R}^{T\cdot n_y \times n_{x,i}}$ the observability matrix and $R_i \in \mathbb{R}^{T\cdot n_y}$ is easily obtained from (12). Therefore, we estimate $\tilde{\mathbf{x}}_i$ by replacing $\mathbf{z}_i$ with $\mathbf{z}_{PM}$ in (13) and then solving for $\tilde{\mathbf{x}}_i$. This gives:

$$\tilde{\mathbf{x}}_i^{\text{es}}(k-T) = \left(\mathcal{O}_i^\top\mathcal{O}_i\right)^{-1}\mathcal{O}_i^\top\left(\mathbf{z}_{PM}(k-1) - R_i\right), \tag{14}$$

provided that $\mathcal{O}_i^\top\mathcal{O}_i$ is invertible, which allows one to obtain $\tilde{\mathbf{x}}_i^{\text{es}}(k)$ as:

$$\tilde{\mathbf{x}}_i^{\text{es}}(k) = A_i^T\tilde{\mathbf{x}}_i^{\text{es}}(k-T) + \sum_{j=1}^{T} A_i^{j-1}B_i\tilde{\mathbf{u}}_i(k-j). \tag{15}$$

The estimated state $\tilde{\mathbf{x}}_i^{\text{es}}(k)$ is then used to obtain the next state $\tilde{\mathbf{x}}_i(k+1)$ following the update equation:

$$\begin{aligned}\tilde{\mathbf{x}}_i(k+1) = A_i((1-\omega_i(\mathbf{u}(k)))\tilde{\mathbf{x}}_i^{\text{es}}(k) \\ + \omega_i(\mathbf{u}(k))\tilde{\mathbf{x}}_i(k)) + B_i\tilde{\mathbf{u}}_i(k).\end{aligned} \tag{16}$$

Notice that, for weighting functions that can take any values between 0 and 1, two linear submodels may have weights equal to 0.5. In that case, there is no clear *active* linear submodel, rather two *partially active* linear models. Hence, equation (16) might badly impact the overall dynamics of the polytopic model. For that reason, we perform the state update (16) if the weight associated to one specific linear submodel is below a threshold value. For the rest of this paper, the threshold value is set to 0.1, *i.e.*, if the linear model participates to only 10% of the global model output, its state is updated using equation (16).

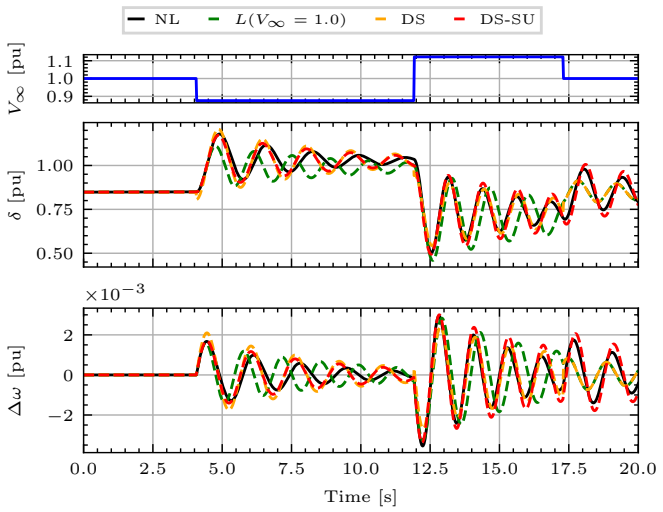


Fig. 4: Comparison between a linear model and polytopic models with and without the modified state update technique.

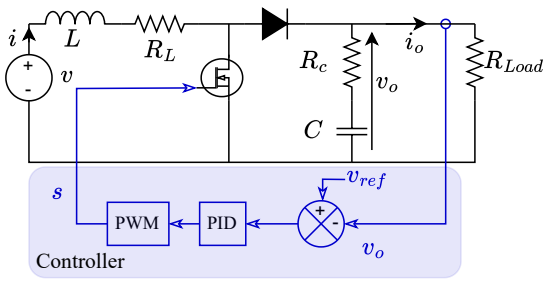


Fig. 5: Voltage-regulated DC-DC boost converter.

In Fig. 3, we show the state trajectories in a phase portrait of the polytopic model with the PWA weighting function with and without the proposed state update technique. One can observe that the state update technique prevents "jumps" from happening. We then consider the same piecewise constant signal for the input as in Fig. 2. The R^2 scores for the linear model, the DS polytopic model and the DS polytopic model with state update are [0.673,0.047], [0.924,0.768], [0.963,0.891], respectively. The simulation results are shown in Fig. 4. The modified state update technique helped improve the performance of the polytopic model.

IV. DC-DC BOOST CONVERTER

Considered voltage-regulated DC-DC boost converter is illustrated in Fig. 5.

The Pulse-Width Modulation (PWM) block produces a binary signal s that triggers the switch. We consider an ideal switch and an ideal diode, *i.e.*, there are no losses or forward voltage drop. We also do not consider saturation of the magnetic cores. When the switch is open, the diode is conducting and the governing dynamical equations for the boost converter are equations (17). On the other hand, when the switch is closed, the dynamical equations are equations (18).

$$\frac{di}{dt} = \frac{v - v_o - R_L i}{L} \tag{17a}$$

$$\frac{dv_o}{dt} = \frac{1}{C(1 + R_c/R_{Load})} \left(\left(\frac{-1}{R_{Load}} - R_c \frac{C}{L} \right) v_o + \left(1 - R_c R_L \frac{C}{L} \right) i + R_c \frac{C}{L} v \right) \tag{17b}$$

$$\frac{di}{dt} = \frac{v - R_L i}{L} \tag{18a}$$

$$\frac{dv_o}{dt} = \frac{-v_o}{C\left(R_{Load} + R_c\right)} \tag{18b}$$

Let us consider the inputs $\mathbf{u} = [v, R_{Load}]^\top$ and outputs $\mathbf{y} = [i, v_o]^\top$. In the following, we assume that $v \in [20, 30]$ V and $R_{Load} \in [20, 50]$ Ω. Notice that the power converter switching model is capable of representing ripples. A major advantage of our black-box model is that it can be run with a larger time step, leading to less computational burden. One drawback, however, is that averaged model cannot represent high frequency dynamics of the converter, *i.e.*, it cannot represent voltage and current ripples, for instance.

A. Linear model identification

We identify four linear models based on an orthotope partitioning of the operating space, *i.e.*, $u_1 = [22.5, 27.5], u_2 = [22.5, 42.5], u_3 = [27.5, 27.5], u_4 = [27.5, 42.5]$. For each setpoint $u_i \ \forall i \in \{1, ..., 4\}$, we design an experiment to gather some measurements on the converter. The experiments are realized with pseudo-random binary sequences.

Notice that in this paper, we simulate in PYTHON the power converter using the switching equations (17)-(18). A digital implementation of the PID controller as well as the PWM are responsible for producing the signal s, which selects whether equation (17) or equation (18) to simulate the dynamical system. Notice that the duty cycle d is the average value of the binary signal s over a switching period.

Remark IV.1 (Practical linear model identification)**.** *We simulate the switching equations of a power converter, and use the simulation results as artificial measurements. However, the methodology proposed to identify the linear submodels can easily be reproduced on a real power converter. The steps in the input voltage can be performed with a controllable DC voltage source, while the steps in the load resistance can be performed by connecting and disconnecting a parallel resistor to the converter. In practice, the dynamics of external devices such as voltage source or sensors may interfere when taking measurements to identify the system. However, there exist techniques to remove undesired dynamics by post-processing measurements [22].*

After obtaining the four different linear models, we use a PWA weighting function to design our polytopic model. We compare the performances of a linear model and our polytopic model in Fig. 6. The $R^2 = [R^2_{v_o}, R^2_i]$ scores are $[0.794, 0.848]$ and $[0.727, 0.652]$ for the polytopic model with PWA weighting function and the linear model, respectively One can notice some steady-state errors in the polytopic model, which emphasizes the importance of the weighting function. In the next section, we introduce a neural network-based weighting function to optimally combine the responses of the different linear models.

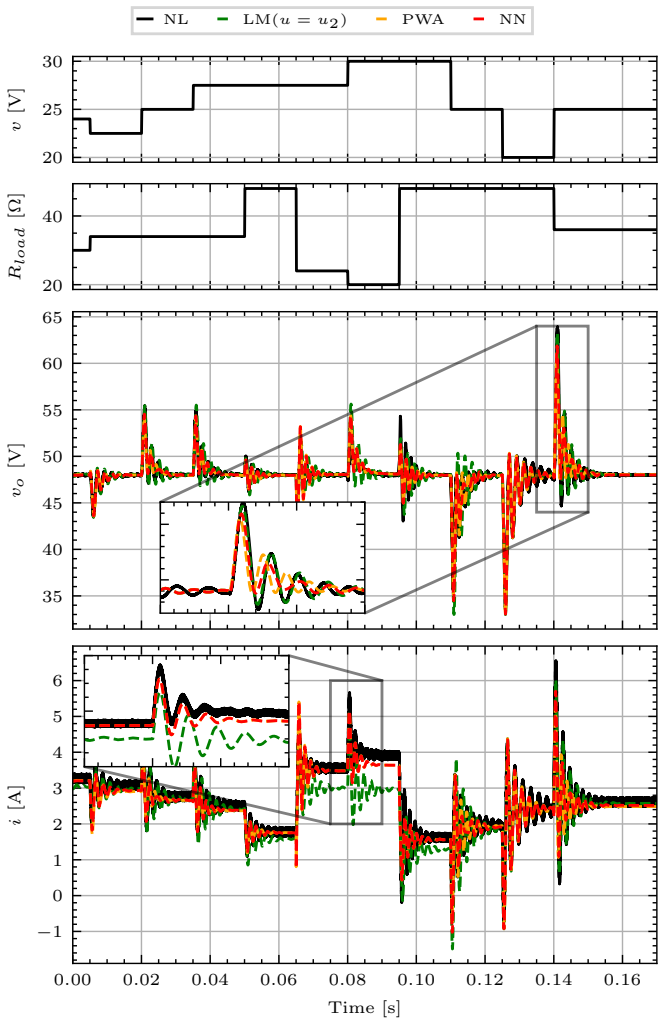


Fig. 6: Performance comparison between polytopic models with PWA and NN weighting functions and a linear model.

B. Neural network-based weighting functions

Figure 7 shows the overall architecture of the proposed polytopic model (*PM-net*) including the neural-network based weighting functions (*NN-WF*). It is composed of one dense neural network for each input v_j that is called a premise variable. For every discrete time k, *NN WF* takes as input the value of $\mathbf{v}(k) = [v_1(k), ..., v_{n_u}(k)]^\top$ and returns a weight value $\omega_i(k)$ comprised in $[0, 1]$ for each linear submodel $i \in \mathcal{N}$. The premise variables $\mathbf{v}$ may differ from the actual system inputs $\mathbf{u}$ if one wants to normalize the inputs fed to the neural network through a function $f(\cdot) : \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_u}$. The output of *NN WF* corresponding to each input $v_j(k)$ is $\boldsymbol{\beta}_j(k) = [\beta_j^1(k), ..., \beta_j^N(k)]^\top \in \mathbb{R}^N$, each element corresponding to one linear submodel. The end layer of each neural network is a *Softmax* function such that the weights for each submodel $i \in \mathcal{N}$ for one specific input $v_j(k)$ sum to one:

$$\sum_{i \in \mathcal{N}} \beta_j^i(k) = 1. \tag{19}$$

The non-normalized weight corresponding to linear submodel i is computed as follows:

$$\gamma_i(k) = \prod_{j=1}^{n_u} \beta_j^i(k). \tag{20}$$

Then the weight associated with each linear submodel i is computed as:

$$\omega_i(k) = \frac{\gamma_i(k)}{\sum_{i \in \mathcal{N}} \gamma_i(k)}. \tag{21}$$

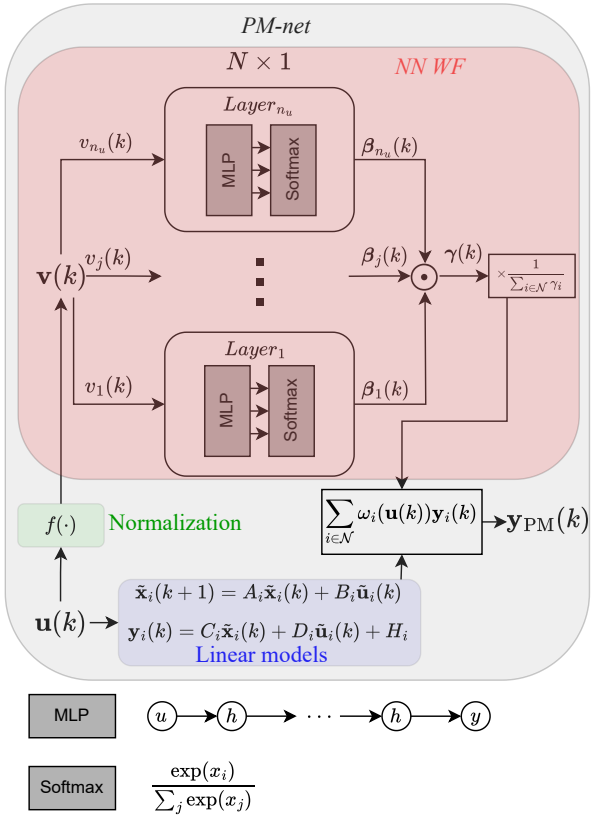
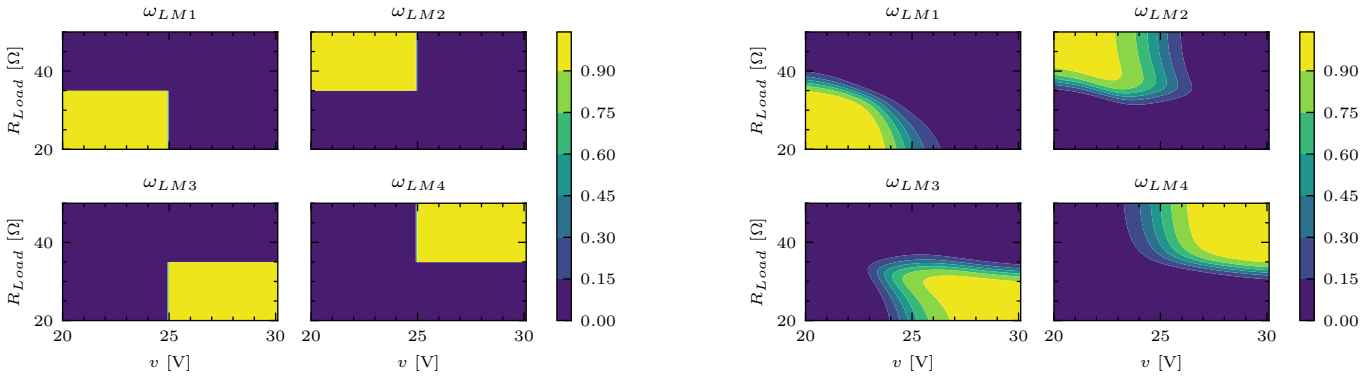


Fig. 7: *NN WF* topology and its interconnection with the linear submodels.

The neural network is trained using measurements gathered on the converter for inputs ranging in the operating space. That would correspond to a normal operation of the power converter. Notice that the normalization function

$$f(u_i(k)) = v_i(k) = 2\left(\frac{u_i(k) - \min_k u_i(k)}{\max_k u_i(k) - \min_k u_i(k)} - 0.5\right), \tag{22}$$

where the operators $\min_k$ and $\max_k$ correspond to minimum and maximum values over all the samples k, ensures that every input $v_i(k) \in [-1, 1]\ \forall i \in \{1, \cdots, n_u\}$. Also, the MLP layers for each input $v_i(k)$ correspond to a linear layer $\mathbb{R} \mapsto \mathbb{R}^{10}$, an hyperbolic tangent activation function, and another linear layer $\mathbb{R}^{10} \mapsto \mathbb{R}^N$ where N denotes the number of linear submodels. The weights obtained are shown in Fig. 8. The comparison between the polytopic model with the neural network-based weighting functions, the PWA weighting functions and the linear model is shown in Fig. 6. The R^2 testing scores are $[0.887, 0.906]$,$[0.794, 0.848]$ and $[0.727, 0.652]$ for the polytopic models with the NN and PWA weighting functions and the linear model, respectively.



(a) PWA

(b) Neural network-based

Fig. 8: PWA and neural network-based weighting functions.

In the next subsection, we introduce a partitioning procedure to add and/or remove linear submodels to improve the polytopic model accuracy.

C. Partitioning procedure

We propose the methodology shown in Alg. 1 for deriving a model *PM-net* for any kind of non-linear system, although this model structure is particularly adapted to power converters. It takes as input the desired operating space $\mathcal{S}$ on which the resulting black-box model should be valid. We start with an orthotope-based partition and identify the linear submodels on the resulting operating points (the centers of the orthotopes) using specifically designed experiments. Then, we train the neural network over a large dataset that excites the system dynamics over its entire operating space. Finally, we analyze the weighting functions associated with each submodel. The latter is divided into two steps: a pruning and a segregation procedure.

The pruning procedure removes submodels to reduce the overall model complexity. If the weight associated with a submodel is always below a specified threshold α, it can be removed since it does not significantly impact the overall model response. This may happen if two submodels have similar parameters as the system behaves linearly along one input, or if one submodel was badly identified. The threshold value is a tuning parameter to achieve the desired overall model complexity.

If no model can be removed, we enter into the segregation procedure. First, one needs to identify the submodel which performs the worst, so that we can identify an operating region where the overall model response can be improved. For each submodel $i \in \mathcal{N}$, we compute the value of a weighted loss function. Let us consider a mean squared error loss function. The weighted loss L_i associated with submodel i can be written as:

$$L_i = \frac{1}{l}\sum_{k=1}^{l}\left[\omega_i(\mathbf{u}(k))(\mathbf{y}(k) - \mathbf{y}_i(k))\right]^2, \tag{23}$$

with l being the number of measurements and $\mathbf{y}(k)$ the output of the non-linear system. The worst submodel i has the largest weighted loss value, that is

$$i^* = \arg\max_{i\in\mathcal{N}} L_i, \tag{24}$$

where i^* is the index of the worst submodel. In operating regions where the submodel i is not expected to perform well, the weight ω_i is close to 0, and the loss associated does not significantly increase. The weighted loss function is also used in [17], where authors described the Local Linear Model Trees methodology (LOLIMOT). The working principle of LOLIMOT is to analyze every feasible re-partitioning in the operating region associated with the worst model. Then, after identifying the new submodels, they select the best partitioning and continue the procedure. In an n-dimensional operating space, this would imply n re-partitionings and $2\times n$ submodels to be identified. For complex systems with multiple inputs and outputs, the process of identifying new submodels is time-consuming. Achieving a successful identification of a submodel implies the design of new experiments and the measurements of the system response, which requires human-in-the-loop. This part should be minimized to speed up the modeling. Thus, in this work, we analyze the *NN WF* to find one cutting direction that can lead to good partitioning. At every step of the procedure, one must identify only two submodels based on new measurements, which significantly reduces the number of new experiments that have to be carried out on the system.

The cutting direction index j^* is found by looking at the evolution of $\beta_j^{i^*}$ for every $j = 1, ..., n_u$. Through the normalization function $f(\cdot)$, the values of the premise variables v_j are bounded within [-1,1]. Thus, we compute the values of $\beta_j^{i^*}$ for every $j = 1, ..., n_u$ and for values of v_j between -1 and 1, and then numerically compute the gradients. The index of the cutting direction is computed as:

$$j^* = \arg\max_{j=1,\dots,n_u}\left\{\max_{v_j\in[-1,1]}\left|\frac{\partial\beta_j^{i^*}}{\partial v_j}\right|\right\}. \tag{25}$$

Let us consider a multi-input system with a linear behavior with respect to one input u. Two submodels $\mathrm{LM}_1, \mathrm{LM}_2$

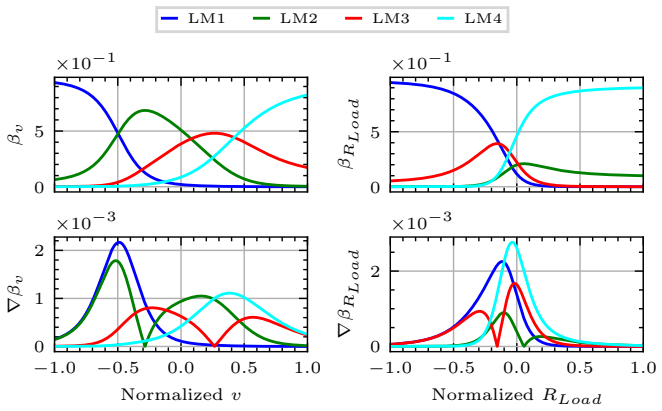


Fig. 9: β values after training the neural network.

identified around two different values of input u have the same parameters. Therefore, the weight associated with LM_1 (β_u^1) or with LM_2 (β_u^2) along various values of u does not vary, as no improvement can be gained by promoting one submodel over the other when u changes. One of the two submodels may be dropped during the pruning procedure. If a system strongly behaves non-linearly along one input, one submodel quickly performs better than the others and the weight associated with that model varies steeply. We thus decide that the best cutting direction corresponds to the input along which the weight varies the steepest.

Let us consider the submodel LM_λ identified around the operating point corresponding to the input $\mathbf{c}^\lambda = [c_1^\lambda, \cdots, c_{n_u}^\lambda]^\top$. We can define an operating region $\mathcal{S}_\lambda \subset \mathcal{S}$ in which the submodel LM_λ is supposed to perform better than the other submodels. Once we found the cutting direction index j^*, we define three new orthotopic regions, $\mathcal{S}_{\lambda-1}$,$\mathcal{S}_\lambda$ and $\mathcal{S}_{\lambda+1}$. We can compute two new operating points, $\mathbf{c}^{\lambda-1}, \mathbf{c}^{\lambda+1}$ from which we can build two perturbation signals to be applied on the system to extract its response. Based on those measurements, we can identify two linear models, $\text{LM}_{\lambda-1}$ and $\text{LM}_{\lambda+1}$, that will be associated with operating regions $\mathcal{S}_{\lambda-1}$ and $\mathcal{S}_{\lambda+1}$. The operating regions for one model i can change from one step to another, as it depends on the number of linear models identified on operating points close to the operating points used to identify model i. The values of β after the first iteration of the method for the DC-DC boost converter are represented in Fig. 9. Following the algorithm, the worst model is the linear submodel 3 (LM3), for which the largest gradient is along R_{Load} direction. The next step is to identify two linear submodels LM5, LM6 around operating points with inputs $(v = 27.5, R_{Load} = 22.5), (v = 27.5, R_{Load} = 32.5)$.

D. Results

The target accuracy ϵ is a R^2 score greater than 0.9 for both output variables. After 6 iterations, we reached $R^2 = [0.907, 0.925]$ with 7 linear submodels. The simulation results are shown in Fig. 10. NN weighting functions better combine the outputs of the different linear submodels. However, they cannot resolve the "jumps" in the system outputs that occur when the input quickly changes. In particular, one can see

Algorithm 1: Operating space partitioning for a non-linear system with n_u inputs. In red, steps that require human-in-the-loop.

input : $\mathcal{S}$
Initialization: $N = 2^{n_u}$, Orthotope partition: $\mathcal{S}_i\ \forall i \in \mathcal{N}$,
Identify LM_i valid in $\mathcal{S}_i\ \forall i \in \mathcal{N}$.
while `Loss` *(PM-net)*$> \epsilon$ **do**
  `Train` (*NN WF*);
  `// Pruning procedure`
  for $i \leftarrow 1$ **to** N **do**
    if $\max(\omega_i) < \alpha$ **then**
      Remove model LM_i;
    end
  end
  if *no model removed* **then**
    `// Segregation procedure`
    $i^* \leftarrow$ from (23),(24);
    $j^* \leftarrow$ from (25);
    Define $\mathcal{S}_{i^*-1}, \mathcal{S}_{i^*+1} \rightarrow$ along j^* direction;
    Identify $\text{LM}_{i^*-1}, \text{LM}_{i^*+1}$;
  end
end

such a "jump" in Fig. 10 around $t = 0.065$ s in the input current i (on the left of the zoomed box). The input current suddenly drops for the polytopic model while it does not for the non-linear model. In Fig. 11, we show how those jumps are removed by appropriately updating the states of the non-active linear models. Notice that, in terms of R^2 score, the modified state update changes the overall performance of the polytopic model ($[0.91, 0.918]$ compared to $[0.907, 0.925]$) but gives a lower RMSE $= \frac{1}{l}\sqrt{\sum_{k=1}^{l} \|\mathbf{y} - \hat{\mathbf{y}}\|^2}$ (l is the number of measurements) value overall (0.454 compared to 0.461).

V. A validation on a small system

We consider a small system composed of a battery, our DC-DC boost converter, and a DC motor, that is illustrated in Fig. 12. The battery is modeled as in [23]:

$$v = \begin{cases} E_0 - Ri - K\frac{Q}{Q - \int idt}(\int idt + i_{\text{filtered}}) + \\ \qquad A\exp(-B\int idt) \quad \text{if discharging,} \\ E_0 - Ri - K\frac{Q}{Q - \int idt}(\int idt) - K\frac{Q}{\int idt - 0.1Q}(i_{\text{filtered}}) \\ \qquad + A\exp(-B\int idt) \quad \text{if charging,} \end{cases} \tag{26}$$

where R is the internal resistance, Q the battery capacity, i_{filtered} the filtered current, $\int idt$ the discharge capacity and E_0, K, A, B parameters that are identified to match a nominal voltage of 48V and a given discharge curve [24]. The DC motor is modeled as in [25]:

$$L_a\frac{di_o}{dt} = v_o - R_a i_o - K_e\omega, \tag{27a}$$

$$J\dot{\omega} = K_e i_o - b\omega - T_L, \tag{27b}$$

where L_a, R_a are the armature inductance and resistance, respectively, K_e is the motor constant, J the motor inertia, b the friction coefficient and T_L the load torque. This system is solved sequentially. The system is initialized with a value for v and R_{Load}. The DC-DC boost converter outputs the input

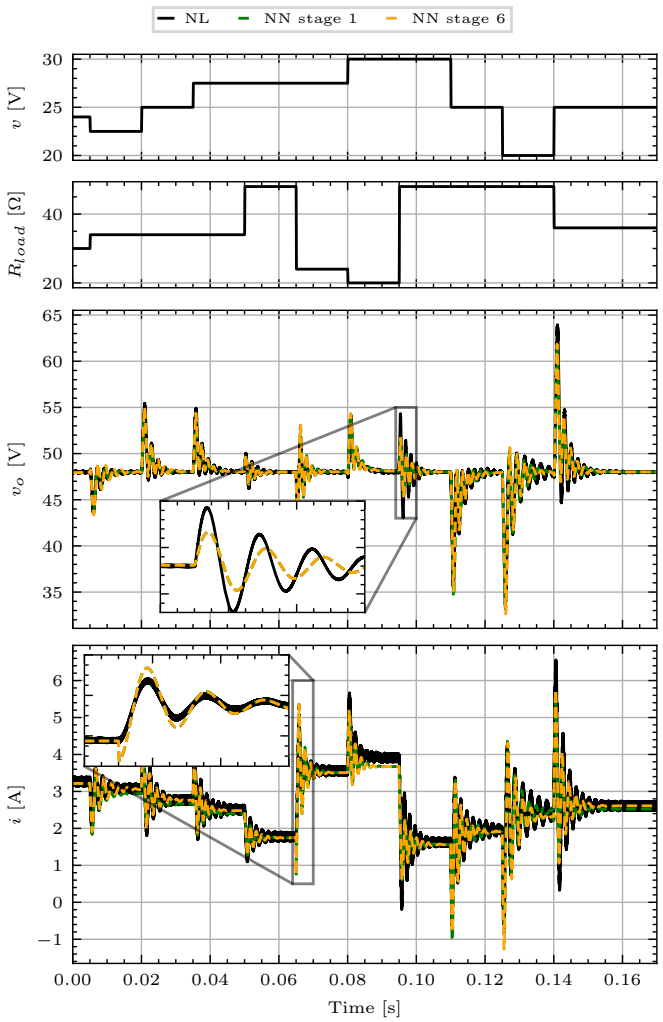


Fig. 10: Performance comparison between two different polytopic models with NN weighting functions at two different stages of Algorithm 1.

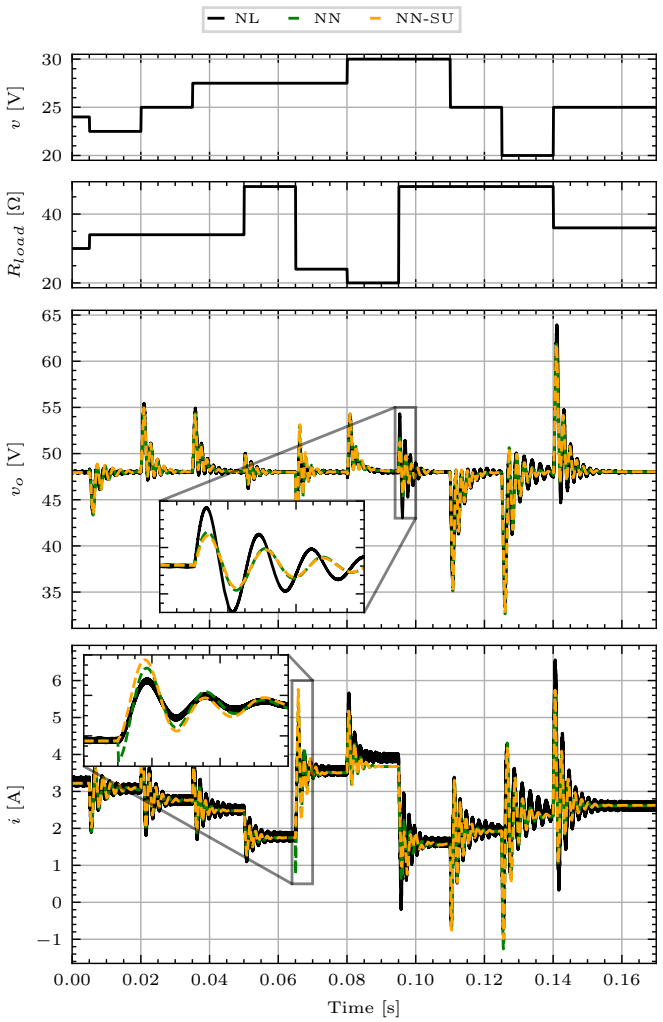


Fig. 11: Performance comparison between two different polytopic models with NN weighting functions with and without modified state update.

current i and the output voltage v_o. The input current i is used in the battery model (26) to obtain the input voltage v, while the output voltage is used in the DC motor model (27) to obtain the output current i_o. One can then build an equivalent resistance $R_{Load} = \frac{v_o}{i_o}$ that is fed into the DC-DC boost converter model. Assuming proper initialization of the state variables, the sequence is described in the following:

$$v_o(k), i(k) = \text{Converter}(v(k), R_{Load}(k)) \tag{28a}$$
$$v(k+1) = \text{Battery}(i(k)) \tag{28b}$$
$$i_o(k+1) = \text{Motor}(v_o(k)) \tag{28c}$$
$$R_{Load}(k+1) = \frac{v_o(k)}{i_o(k+1)} \tag{28d}$$

In Fig. 13, we show the simulation results of the small system, when the DC-DC boost converter is modeled using the switching model, and when it is modeled using our multimodel approach, with and without the modified state update. One can see that our multimodel approach describes accurately the dynamics of the small system. One can also notice that the state update does not improve significantly the dynamical response of the DC-DC boost converter because

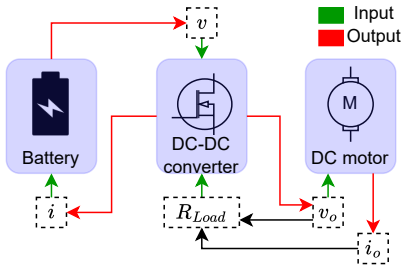


Fig. 12: Illustration of the small system.

there are no "jumps" in the input. The inputs (battery voltage and equivalent resistance obtained from the DC motor) vary smoothly, so that the weights also vary smoothly.

VI. Conclusion

We presented a multimodel approach to derive large-signal black-box models of power electronic converters. The approach is built upon the well-known theory of linear system identification and is consequently simple and mathematically tractable. In this paper, we addressed two important challenges in the multimodel framework; the design of the weighting

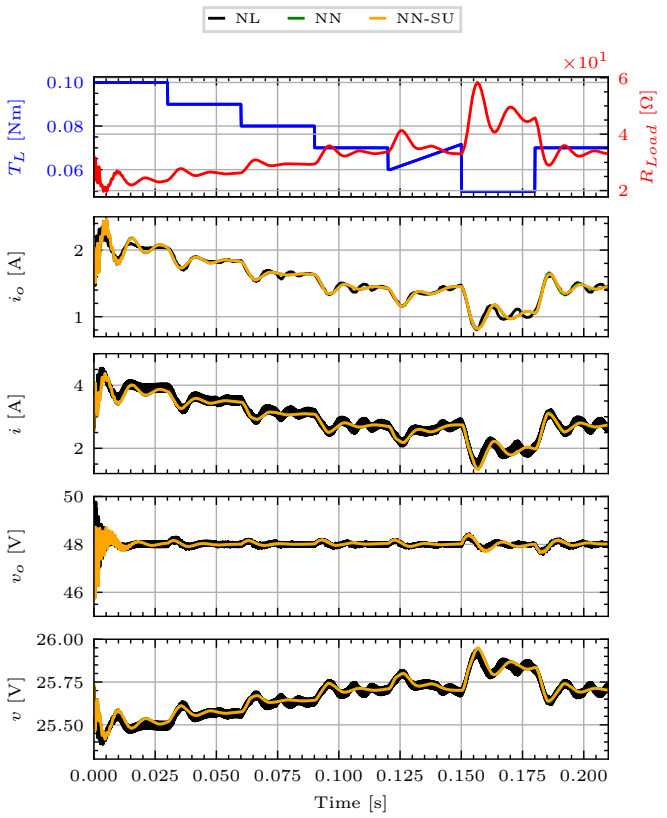


Fig. 13: Performance of polytopic models for system-level analysis.

functions and the partitioning of the operating space. The neural network-based weighting functions offer more flexibility and better combine linear submodels' responses than generic weighting functions. Furthermore, the post-analysis of the trained neural network provides valuable information to improve the performance of the polytopic model by adding or removing some linear submodels. The reliance on linear submodels together with the specific design of the weighting functions renders our multimodel approach interpretable, making it suitable for system-level analysis. The focus of this paper is on detailed presentation of the approach, and it is validated using only a small system comprising a voltage-regulated DC-DC boost converter, a battery and a DC motor. The validation on realistically sized systems and the proposed approach considerations with other types of PECs are central to the research efforts that will be carried out in the future.

References

[1] A. Monti, F. Milano, E. Bompard, and X. Guillaud, *Converter-Based Dynamics and Control of Modern Power Systems*. Academic Press, 2020.

[2] S. Bacha, I. Munteanu, A. I. Bratcu *et al.*, "Power electronic converters modeling and control," *Advanced textbooks in control and signal processing*, vol. 454, p. 454, 2014.

[3] J. D. Lara, R. Henriquez-Auba, D. Ramasubramanian, S. Dhople, D. S. Callaway, and S. Sanders, "Revisiting power systems time-domain simulation methods and models," *IEEE Transactions on Power Systems*, 2023.

[4] M. Paolone, T. Gaunt, X. Guillaud, M. Liserre, S. Meliopoulos, A. Monti, T. Van Cutsem, V. Vittal, and C. Vournas, "Fundamentals of power systems modelling in the presence of converter interfaced generation," *Electric Power Systems Research*, vol. 189, no. 3, p. 106811, 2020.

[5] V. Valdivia, A. Barrado, A. Lazaro, M. Sanz, D. L. del Moral, and C. Raga, "Black-box behavioral modeling and identification of DC-DC converters with input current control for fuel cell power conditioning," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 1891–1903, 2013.

[6] A. Francés, R. Asensi, O. García, R. Prieto, and J. Uceda, "A black-box modeling approach for DC nanogrids," in *2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*. IEEE, 2016, pp. 1624–1631.

[7] L. Arnedo, D. Boroyevich, R. Burgos, and F. Wang, "Polytopic black-box modeling of DC-DC converters," in *2008 IEEE Power Electronics Specialists Conference*. IEEE, 2008, pp. 1015–1021.

[8] S. Chiniforoosh, J. Jatskevich, A. Yazdani, V. Sood, V. Dinavahi, J. A. Martinez, and A. Ramirez, "Definitions and applications of dynamic average models for analysis of power systems," *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2655–2669, 2010.

[9] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[10] A. A. Adeniran and S. El Ferik, "Modeling and identification of nonlinear systems: A review of the multimodel approach—part 1," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1149–1159, 2016.

[11] A. Frances, R. Asensi, O. Garcia, R. Prieto, and J. Uceda, "Modeling electronic power converters in smart DC microgrids—An overview," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6274–6287, 2017.

[12] A. Francés, R. Asensi, O. García, R. Prieto, and J. Uceda, "The performance of polytopic models in smart DC microgrids," in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2016, pp. 1–8.

[13] G. Rojas-Dueñas, J.-R. Riba, and M. Moreno-Eguilaz, "Black-box modeling of DC-DC converters based on wavelet convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[14] G. Rojas-Dueñas, J.-R. Riba, K. Kahalerras, M. Moreno-Eguilaz, A. Kadechkar, and A. Gomez-Pau, "Black-box modelling of a DC-DC buck converter based on a recurrent neural network," in *2020 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2020, pp. 456–461.

[15] S. Sahoo, H. Wang, and F. Blaabjerg, "On the explainability of black box data-driven controllers for power electronic converters," in *2021 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2021, pp. 1366–1372.

[16] M. Al-Greer, M. Armstrong, M. Ahmeid, and D. Giaouris, "Advances on system identification techniques for DC-DC switch mode power converter applications," *IEEE Transactions on Power Electronics*, vol. 34, no. 7, pp. 6973–6990, 2018.

[17] O. Nelles, "Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT)," *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 639–644, 1997.

[18] A. Colot, A. Giannitrapani, S. Paoletti, and B. Cornélusse, "Enhanced neural network-based polytopic model for large-signal black-box modeling of power electronic converters," in *2023 IEEE Belgrade PowerTech*. IEEE, 2023, pp. 01–06.

[19] T. Raz, "ML interpretability: Simple isn't easy," *Studies in History and Philosophy of Science*, vol. 103, pp. 159–167, 2024.

[20] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," *arXiv*, pp. 1–10, 2019.

[21] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient stability of power systems: A unified approach to assessment and control*. Kluwer Academic Publishers, 2000.

[22] I. Cvetkovic, M. Jaksic, D. Boroyevich, P. Mattavelli, F. C. Lee, Z. Shen, S. Ahmed, and D. Dong, "Un-terminated, low-frequency terminal-behavioral dq model of three-phase converters," in *2011 IEEE Energy Conversion Congress and Exposition*. IEEE, 2011, pp. 791–798.

[23] O. Tremblay and L.-A. Dessaint, "Experimental validation of a battery dynamic model for ev applications," *World Electric Vehicle Journal*, vol. 3, no. 2, pp. 289–298, 2009.

[24] A. Colot, B. Bastin, B. Ewbank, F. Frebel, B. Bidaine, and B. Cornélusse, "Hybrid power solution modelling based on artificial intelligence," *IET Conference Proceedings*, vol. 2023, pp. 3114–3119, 2023.

[25] G. C. Konstantopoulos and A. T. Alexandridis, "Enhanced control design of simple DC-DC boost converter-driven DC motors: Analysis and implementation," *Electric Power Components and Systems*, vol. 43, no. 17, pp. 1946–1957, 2015.