Enhanced neural network-based polytopic model for large-signal black-box modeling of power electronic converters

Antonin Colot
Montefiore Institute
University of Liège
Liège, Belgium
antonin.colot@uliege.be

Antonio Giannitrapani
Dpt. of Information Engineering and Mathematics
University of Siena
Siena, Italy

Simone Paoletti
Dpt. of Information Engineering and Mathematics
University of Siena
Siena, Italy

Bertrand Cornélusse
Montefiore Institute
University of Liège
Liège, Belgium

***Abstract*—We propose a large-signal black-box model of power electronic converters inspired by polytopic models. Small-signal models are identified around different operating points to mimic the converter's local dynamics. The linear models' responses are then weighted using a trained neural network to create a large-signal model. The traditional trial and error weighting function tuning of polytopic models can result in a suboptimal combination of linear models. In this work, we use neural networks to approach an optimal combination. The analysis of the trained neural network can enhance the model's accuracy by suggesting new small-signal models. It also permits removing linear models that do not significantly improve the global model's accuracy while reducing complexity. The methodology is applied to a voltage-regulated DC-DC boost converter and provides accurate models of converter dynamics.**

***Index Terms*—polytopic model, power electronic converters, system identification, artificial neural networks**

I. Introduction

Nowadays, distributed energy resources (DERs) are widespread as more renewable energy sources (RESs) are integrated to meet climate target plans, and energy storage systems (ESSs) take an increasing role in dealing with RES intermittency. Modern electrical power systems extensively use power electronic converters (PECs) to interface RES/ESSs with an electrical DC or AC grid (converter-interfaced generation/energy storage systems). PECs optimize the energy yield by maximum power point tracking. Many PEC topologies exist, and their control algorithms differ largely from manufacturer to manufacturer. As a result, PECs have very different behaviors, and unstable electric phenomena may occur when interconnected. There is a need in the industry to model PECs and simulate their interconnection before implementation, especially when designing system-level control. PEC modelization is difficult, as manufacturers tend to restrict the information they disclose. Black-box models are thus investigated, as they are identified based only on measurements and may accurately simulate PEC dynamics. Black-box models can be significantly simpler than detailed models and, therefore, less computationally expensive, which makes them good candidates for real-time simulation. Because PECs are non-linear and time-varying systems, the identification process is complicated. Usually, one uses averaging techniques and linearization around a specific operating point to extract a linear and time-invariant system. Unfortunately, the linearization around an operating point is too restrictive when considering the interconnection of PECs in an electrical system. For instance, in microgrids, PEC flexibilities provide ancillary services such as voltage support. Furthermore, as PECs are interfaced with intermittent renewable energy sources, a linear model may no longer be valid since the operating point often changes [1].

Several techniques have been proposed for developing large-signal black-box models of PECs. The most used technique is the identification of a polytopic model that weights the responses of linear models defined around different operating points [2]. Polytopic models can then be used for local stability analysis [2] or for system-level analysis [3], [4]. However, they have two main drawbacks [3]; the identification of the weighting functions of the linear models is not automated, and the ideal partitioning of the PEC operating space is a priori unknown. In the literature, a double sigmoid is usually used as a weighting function to ensure a smooth transition between models while keeping a simple mathematical expression [5]. To tackle the second issue, [6] proposes to analyze how the different identified frequency response functions vary to partition the PEC operating space in smaller regions (more linear models in non-linear regions). Nevertheless, there does not exist automated way to construct those polytopic models. Trial-and-error methods are necessary to identify the parameters of the weighting functions and partition the operating space optimally. Other techniques use deep neural networks [7], [8] and do not require the identification of linear models or some weighting functions. Using recurrent neural networks, those methods can accurately represent PEC dynamics over a large operating space. Nevertheless, they need a large amount of data, and the model performance depends on the measurements. For instance, the dynamics of the measurement/external electrical devices, such as voltage sources, and current sinks, can interfere with the dynamics of the PEC. This will be

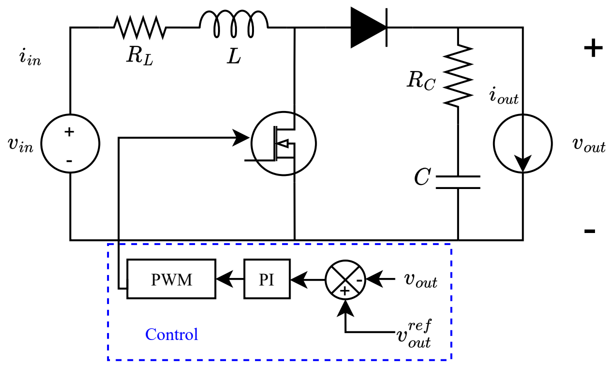

Fig. 1. Voltage regulated DC-DC boost converter using a proportional-integral (PI) controller. The load is a current sink.

reflected in the measurements and eventually in the resulting model.

We propose a new methodology to develop large-signal black-box models of PECs. The main contributions are:

- A neural network (NN)-based polytopic model in which the NN optimally combines the responses of *linear models* is introduced.
- The trained NN are used to improve the partitioning of the PEC operating space. The analysis of the trained NN gives useful information about which model can be dropped or where to add new linear models.

The paper is organized as follows. Section II presents modeling techniques for PECs based on linear models (*G-parameters models*) and polytopic models. Section III introduces the proposed methodology and compares its performance on a toy example with methods based on polytopic models. Section IV reports results on a voltage-regulated DC-DC boost converter. Section V concludes the paper.

II. Review of PEC modeling techniques

A. G-parameters model

The *G-parameters model* is a two-port representation of a converter. This method is suitable for analyzing PEC interactions in large systems, as one can easily combine different subsystems to model different architectures [1]. Each block in the two-port representation corresponds to a transfer function, mapping one input to an output. If a detailed converter model is available, the transfer functions are built based on an averaged model (replicating the average behavior of the system dynamics). Then, a linearized model around an operating point is constructed. This model is only valid for slight variations around the considered operating point. The form of the *G-parameters model* for a voltage-regulated DC-DC boost converter (whose schematic is shown in Fig. 1) is given by:

$$\begin{pmatrix} \tilde{v}_{out} \\ \tilde{i}_{in} \end{pmatrix} = \begin{pmatrix} G(s) & Z(s) & G_{v-vref}(s) \\ Y(s) & H(s) & G_{i-vref}(s) \end{pmatrix} \begin{pmatrix} \tilde{v}_{in} \\ \tilde{i}_{out} \\ \tilde{v}_{out}^{ref} \end{pmatrix} \quad (1)$$

The control loop that sets the duty cycle, as well as the modulator that provides the PWM signal, are included in the two-port model ($G_{v-vref}(s), G_{i-vref}(s)$). The inputs of the *G-parameters model* are small deviations from a specified operating point, *i.e.* $\tilde{v}_{in} = v_{in} - V_{in}$ where V_{in} is the value of the input voltage for the specified operating point. v_{in} corresponds to the input DC voltage, v_{out} is the output DC voltage and v_{out}^{ref} the reference setpoint for the output DC voltage. i_{in} is the input current also called the inductor current, and i_{out} corresponds to the load current. In steady-state, $v_{out} = v_{out}^{ref}$ if there is no saturation and $i_{in} = \frac{v_{out}\, i_{out}}{\eta\, v_{in}}$, where η is the converter efficiency, which depends on the operating point. Without a detailed model, the transfer functions can be estimated from measurements. Each transfer function is defined independently, applying a small disturbance on one input while keeping the others constant during the measurement process. The dynamics of other voltage/current sources or loads should not interfere. In practice, it is difficult to meet this constraint. Thus, techniques to remove those dynamics have been investigated [4], [9].

B. Polytopic models

Polytopic models combine several linear models' responses using weighting functions to improve *G-parameters models* in the case of a non-linear system. Suppose that one wants to approximate a non-linear model $\mathbf{F}(\mathbf{x})$. Let $\mathbf{G}_i(\mathbf{x})$ be a linear model that approximates $\mathbf{F}(\mathbf{x})$ for values of $\mathbf{x}$ around $\mathbf{x}_i$. Let $\omega_i(\mathbf{x})$ be the weighting function associated to the linear model $\mathbf{G}_i(\mathbf{x})$. For each value of $\mathbf{x}$, $\omega_i(\mathbf{x})$ takes a value in $[0, 1]$. It indicates the confidence that we give to the model $\mathbf{G}_i(\mathbf{x})$ as an estimate of the non-linear model $\mathbf{F}(\mathbf{x})$ at the point $\mathbf{x}$. For instance, it makes sense that $\omega_i(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_i} = 1$ as $\mathbf{G}_i(\mathbf{x})$ has been identified at $\mathbf{x}_i$. The polytopic model can be written as

$$\hat{\mathbf{F}}(\mathbf{x}) = \sum_{i=1}^{N} \omega_i(\mathbf{x})\mathbf{G}_i(\mathbf{x}), \quad (2)$$

where N is the number of linear models considered and $\hat{\mathbf{F}}(\mathbf{x})$ corresponds to an estimate of the non-linear model $\mathbf{F}(\mathbf{x})$. For polytopic models, we have the following constraint:

$$\forall \mathbf{x} \in \mathcal{O} \subset \mathcal{R}^{n_x}, \sum_{i=1}^{N} \omega_i(\mathbf{x}) = 1, \quad (3)$$

where $\mathcal{O}$ is the operating space, and $\omega_i(\mathbf{x})$ is the weight multiplying the response of linear model i for a given input $\mathbf{x}$. This constraint expresses that the sum of linear models' responses is not attenuated nor amplified. The constraint removes degrees of freedom, as the slopes of two adjacent weighting functions (as well as their centers) must be equal [5]. Polytopic models ensure that the best-fitted linear model is responsible for most of the model response as we span the system operating space. In the case of non-linear systems, it outperforms linear models but has two major drawbacks: there is no automated way to identify the weighting functions and the ideal partitioning of the PEC operating space is a priori unknown.

III. Proposed enhanced polytopic model

A. Neural network-based polytopic model: PM-net

We propose a NN method to resolve the main drawbacks of polytopic models. The method includes:

- Weighting functions modeled by a neural network with parameters tuned using a gradient descent algorithm.
- A methodology to improve the partitioning of the PEC operating space based on the neural network analysis.

The NN used to provide weights for each linear model is referred to as *NN WF* in the following and is shown in Fig. 2 along *PM-net*. It is composed of one multi-layer perceptron (MLP) for each input $\mathbf{v}_j$ that is called a premise variable. The premise variables may differ from the actual inputs of the linear models $\mathbf{x}$ if one wants to normalize the inputs fed to the neural network. The end layer of each MLP is a $Softmax$ function such that the weights for each model $i = 1, ..., N$ for one specific input $\mathbf{v}_j$ sum to one:

$$\sum_{i=1}^{N} \beta_i(\mathbf{v}_j) = 1, \tag{4}$$

with N the number of linear models. The output of *NN WF* corresponding to each input $\mathbf{v}_j$ for model i are then combined as:

$$\gamma_i = \prod_{j=1}^{n_x} \beta_i(\mathbf{v}_j). \tag{5}$$

Then the weight associated with each model i is computed as:

$$\omega_i = \frac{\gamma_i}{\sum_{i=1}^{N} \gamma_i}. \tag{6}$$

This ensures that constraint (3) is satisfied. We propose the methodology shown in **Algorithm 1** for partitioning the PEC operating space. It takes as input the desired operating space on which the resulting black-box model should be valid. We start with an orthotope-based partition and identify the linear models on the resulting operating points (the centers of the orthotopes) using specifically designed measurements. Then we train the neural network over a large dataset that includes the system dynamics over its entire operating space. Finally, we analyze the weighting functions associated with each linear model. The latter is divided into two steps: a pruning and a segregation procedure.

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

If no model can be removed, we enter into the segregation procedure. First, one needs to identify the linear model G_{worst}, which performs the worst, so that we can identify an operating region where the overall model response can be improved. For each linear model i, we compute the value of a weighted loss

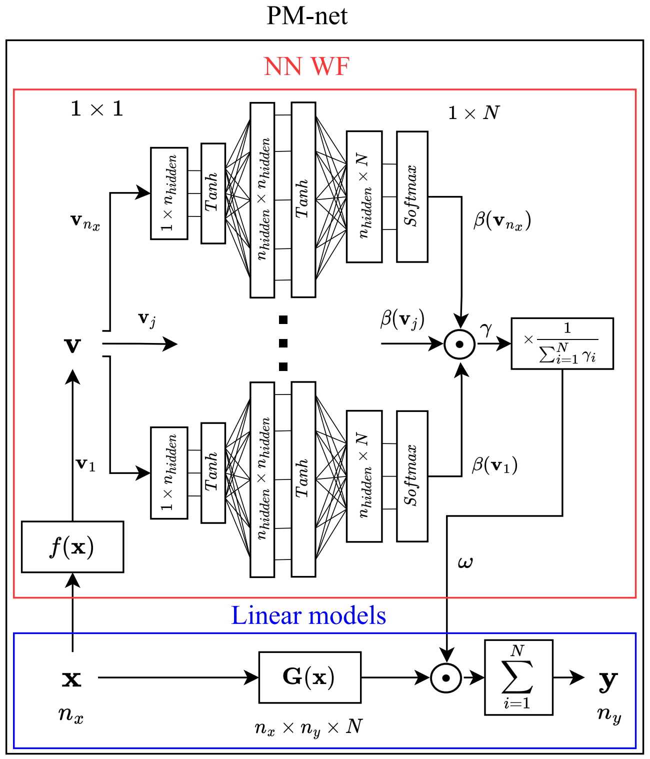

Fig. 2. *NN WF* topology and how it is interconnected with linear models $\mathbf{M}(\mathbf{x})$. For each time t, *NN WF* takes as input the value of $\mathbf{v}$ at time t and returns a weight value $\omega_i(\mathbf{v}[t])$ comprised in $[0, 1]$ for each linear model in $\mathbf{M}(\mathbf{x})$. For each time t, the output of each linear model is computed and then multiplied by the corresponding weight.

function. Let us consider a mean squared error loss function. The weighted loss L_i associated with linear model $\mathbf{G}_i$ can be written as:

$$L_i = \frac{1}{l} \sum_{k=1}^{l} \left(\omega_i(\mathbf{x}[k])(\mathbf{F}(\mathbf{x}[k]) - \mathbf{G}_i(\mathbf{x}[k]))\right)^2, \tag{7}$$

with l the number of measurements. The worst linear model $\mathbf{G}_{worst}$ has the largest weighted loss value, that is $G_{worst} = G_{i^*}$ where

$$i^* = \arg \max_{i=1,\dots,N} (L_i). \tag{8}$$

In operating regions where the linear model 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 [10], 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 linear models, they select the best partitioning and continue the procedure. In an n-dimensional operating space, this would imply n re-partitioning and $2 \times n$ linear models to be identified. For PECs modeling, the process of identifying new linear models is time-consuming. Achieving a successful identification of a linear model implies the design of new experiments and the measurements of the converter 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 linear models based on new measurements, which significantly reduces the number of new experiments that have to be carried out on the converter. The cutting direction $\mathbf{v}_{j^*}$ is found by looking at the gradients of the weighting functions $\beta_{i^*}(\mathbf{v}_j)$ associated with the worst linear model as:

$$j^* = \arg\max_j \left\{ \max_k \left| \frac{\partial \beta_{i^*}}{\partial \mathbf{v}_j} \right|_{\mathbf{v}_j = \mathbf{v}_j[k]} \right\}. \tag{9}$$

The gradient of the weighting functions $\beta_{i^*}(\mathbf{v}_j)$ for every input $\mathbf{v}_j$ boils down to a vector of length n_x (the number of inputs). Let us consider a multi-input system with a linear behavior with respect to one input u. Two linear models G^1, G^2 identified around two different values of input u have the same parameters. Therefore, the weight associated with G^1 ($\beta_1(u)$) or with G^2 ($\beta_2(u)$) along various values of u does not vary, as no improvement can be gained by promoting one linear model over the other when u changes. One of the two linear models may be dropped during the pruning procedure. On the contrary, if a system strongly behaves non-linearly along one input, one linear model 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 linear model $\mathbf{G}_{worst}$ identified around the operating point $\mathbf{x}_{worst}$. We can define an operating region $\mathcal{O}_{worst} \subset \mathcal{O}$ in which the linear model $\mathbf{G}_{worst}$ is supposed to perform better than the other linear models. Once we found the cutting direction $\mathbf{v}_{j^*}$, we define three new orthotopic regions, $\mathcal{O}^-_{worst}$,$\mathcal{O}^0_{worst}$ and $\mathcal{O}^+_{worst}$. We can compute two new operating points, $\mathbf{x}^-_{worst}, \mathbf{x}^+_{worst}$ 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, $\mathbf{G}^-_{worst}$ and $\mathbf{G}^+_{worst}$, that will be associated with operating regions $\mathcal{O}^-_{worst}$ and $\mathcal{O}^+_{worst}$. 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.

Finally, we rerun the algorithm until reaching a target accuracy ϵ.

B. Performance comparison on a toy example

We use the non-linear system (10) as an example to compare performance of classic polytopic models and *PM-net* at estimating the behavior of a non-linear system.

$$\begin{aligned} y[k] =& 0.95y[k-1] - 0.5p[k-1] \\ &+ 0.1u[k] + 0.5\arctan(u[k-2]^2) \end{aligned} \tag{10}$$

Assuming we want our model to be valid on $p \in [-2, 2], u \in [-5, 5]$, we identified 4 linear models $\mathbf{G}_i(p, u)$ at points $\mathbf{x}_i \in \{[-1, -2.5], [1, -2.5], [-1, 2.5], [1, 2.5]\}$. The linear models

Algorithm 1: Operating space partitioning for a n-dimensional non-linear system. In red, requires human-in-the-loop.

```
input: $\mathcal{O}$
// Split $\mathcal{O}$ into $2^n$ orthotropic regions
$\mathcal{O}_i$ = split_OS($\mathcal{O}$);
// Identify linear models
for $i \leftarrow 1$ to $N$ do
    $\mathbf{x}_i$ = select_point ($\mathcal{O}_i$);
    $\mathbf{G}_i$ = LM_id ($\mathbf{x}_i$);
end
// Initialize PM-NET
while Loss (PM-NET)> $\epsilon$ do
    Train (PM-NET);
    // Pruning procedure
    for $i \leftarrow 1$ to $N$ do
        $\omega_i$ = NN_WF (PM-NET,i);
        if max($\omega_i$) < $\alpha$ then
            drop_model ($\mathbf{G}_i$);
            Train (PM-NET);
        end
    end
    // Segregation procedure
    $G_{worst} \leftarrow$ (7),(8);
    $\mathbf{v}_{j^*} \leftarrow$ (9);
    $\mathbf{x}^-_{worst}, \mathbf{x}^+_{worst}$ = select_point ($\mathcal{O}^-_{worst}, \mathcal{O}^+_{worst}$);
    $\mathbf{G}^-_{worst}, \mathbf{G}^+_{worst}$ = LM_id ($\mathbf{x}^-_{worst}, \mathbf{x}^+_{worst}$);
end
```

are identified based on a small-signal analysis *i.e.* we perturb the system around one operating point and extract its response. The system is perturbed one input at a time such that we build a *G-parameters model* as a local estimate of the system. The studied system is linear with respect to p, but non-linear with respect to u. One can derive a linear surrogate, valid around operating point $\mathbf{x}_i = [u_i, p_i]^T$ such as:

$$y[k] = \frac{1}{1 - 0.95z^{-1}} \begin{pmatrix} 0.1 + 0.5f(u_i)z^{-2} \\ -0.5z^{-1} \end{pmatrix} \begin{pmatrix} u[k] & p[k] \end{pmatrix}, \tag{11}$$

with

$$f(u_i) = \frac{2u_i}{1 + (u_i)^4}. \tag{12}$$

We study the performance of two weighting functions, a triangular function, and a double sigmoid. The double sigmoid is a common choice for black-box models of PECs, while triangular weighting functions have been used when considering fuzzy models such as the one defined by Takagi-Sugeno [11]. The double sigmoid function is

$$\begin{aligned} \omega_i(\mathbf{x}) = & \left(\frac{1}{1 + \exp\left(-s^p_1 \left(p - (p_i - 1)\right)\right)} \right. \\ & \left. - \frac{1}{1 + \exp\left(-s^p_2 \left(p - (p_i + 1)\right)\right)} \right) \\ & \left(\frac{1}{1 + \exp\left(-s^u_1 \left(u - (u_i - 2.5)\right)\right)} \right. \\ & \left. - \frac{1}{1 + \exp\left(-s^u_2 \left(u - (u_i + 2.5)\right)\right)} \right). \end{aligned} \tag{13}$$

It has two degrees of freedom, s_1 and s_2, for each input. Both have to be positive and large enough such that $\omega_i(\mathbf{x})$ can reach the unitary value at $\mathbf{x} = \mathbf{x}_i$. Finally, we also trained *NN WF* to combine the linear models. The resulting *G-parameters models* are included in a *dynoNet* topology [12] along MLPs for the weighting function. We used a *dynoNet* topology because it implements analytic formulas for differentiating transfer functions defined in the *z-domain*. Fig. 3 illustrates **Algorithm 1**. In the first iteration, the operating space $\mathcal{O}$ is split into four operating regions. Two of the four models can be dropped as the system behaves linearly along p. In the third iteration, we train the *NN WF* and compute the weighted loss L_i for the two remaining linear models. The operating region associated with the worst performing model (see (8)) is re-partitioned. We analyze the gradient of the weighting function to identify the cutting direction, see (9), and observe that the gradient along u is the largest. Two new linear models are added to enhance the model's performance, and new operating regions are defined. The algorithm stops after six iterations, the required accuracy being reached.

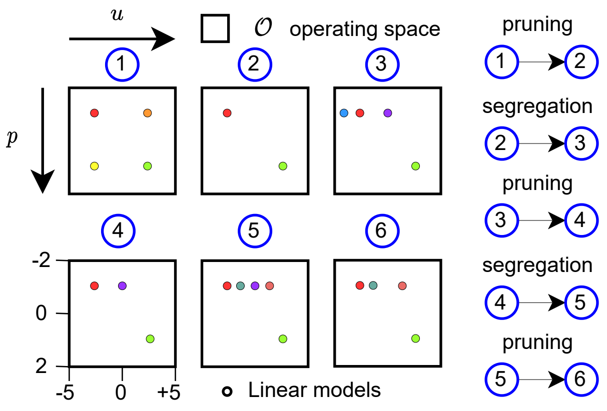

Fig. 3. Results after 6 iterations of **Algorithm 1**.

Fig. 4 compares the performance of the polytopic models with different weighting functions (triangular: $PM-\Delta$, double sigmoid:$PM-S$ and *PM-net*). The *PM-net* model performs much better than the other techniques for the same number of linear models, even though no information about the identification process of the linear models or the structure of the non-linear system has been given to *NN WF*.

IV. Application to a DC-DC boost converter

We analyze a voltage-regulated DC-DC boost converter (see Fig. 1), with a simple topology, but dynamics that strongly depend on the operating point [5]. We are using a detailed model of the converter which provides us a response close to reality.

A. Operating space and first partitioning

The DC-DC boost converter has three inputs and two outputs. The inputs and outputs of the *G-parameters models* are $\mathbf{x} = [v_{in}, p_{out}, k]$ and $\mathbf{y} = [v_{out}, i_{in}]$, respectively. We considered the output to input voltage ratio $k = \frac{v_{out}^{ref}}{v_{in}}$ and

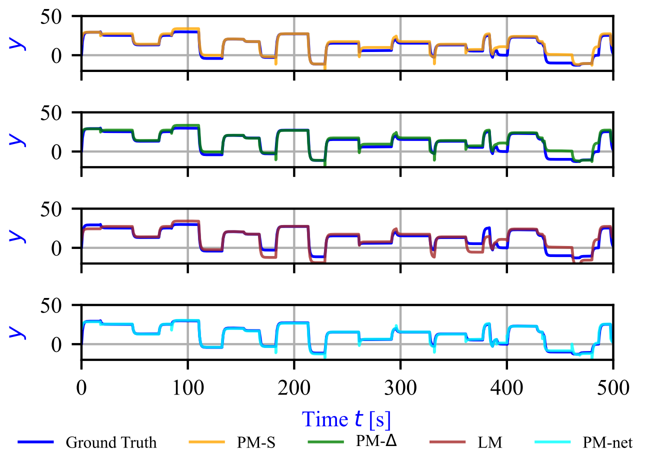

Fig. 4. Performance of the polytopic models and a linear model. The mean squared error (MSE) is used to compare their performance at estimating the non-linear system (10). The system is simulated for 500 seconds, with a sampling frequency of 20Hz, and the MSE is computed based on the error between the actual output of the system and the estimate at those points. MSE for LM: 1.5264, $PM-\Delta$: 0.9426, $PM-S$: 0.9210, *PM-net*: 0.1060

TABLE I
DC-DC BOOST CONVERTER PARAMETERS.

Inductor L	1 mH
Inductor losses R_L	0.1 Ω
Capacitor C	810 μF
Equivalent series resistor R_C	0.01 Ω
Switching frequency f_s	50 kHz
Proportional term P	0.006
Integral term I	0.4364

the load power $p_{out} = v_{out}^{ref}\ i_{out}$ to highlight the dynamic dependency on the operating point. Indeed, k depends non-linearly on the converter duty cycle, and the converter efficiency (which impacts i_{in}) is a non-linear function of p_{out} and v_{in}. The inputs $\mathbf{x}$ of the *G-parameters models* are normalized to be bounded in $[-1, 1]$, and the resulting $\mathbf{v}$ is fed to *NN WF*. The operating space considered is $v_{in} \in [30, 40]V, p_{out} \in [10, 210]W, k \in [1, 2]$. Following **Algorithm 1**, we define linear models in 2^3 orthotopic regions.

B. Identification of linear models

The linear models are *G-parameters models*, and each transfer function composing the *G-parameters model* (1) is identified by setting all the values of $\mathbf{x}$ to zero except one where we apply a small-disturbance. The small disturbance corresponds to a pseudo-random binary sequence (PRBS), and the transfer functions are further validated using a different PRBS. Each transfer function corresponds to Output-Error (OE) models. The parameters identification of the OE models depends on the user choice. In this work, we use the function OE in MATLAB. The order of each transfer function is then reduced using the function BALRED based on the Hankel singular values, *i.e.* states with low energy are dropped.

C. Dataset and metric used

The dataset contains the converter's dynamics over its entire operating space. We have measured the converter's response over 200 seconds of operation for the training dataset. We also have validation, and testing datasets, which contains the converter's response over 60 seconds of operation. The loss function used to train the *NN WF* is

$$Loss = \frac{1}{l}\sum_{i=0}^{l}\left(\left(\frac{v_{out,i} - \hat{v}_{out,i}}{V_{base}}\right)^2 + \left(\frac{i_{in,i} - \hat{i}_{in,i}}{I_{base}}\right)^2\right), \quad (14)$$

where V_{base} and I_{base} correspond to base values defined as the maximum value of $v_{out,i}$ and $i_{in,i}$ calculated over the number of measurements l of the dataset, respectively, where $i = 0, .., l$ is the index of the measurement.

D. Topology of the NN WF and training procedure

The *NN WF* used is composed of one hidden layer. For the DC-DC boost converter, we have $n_x = 3$, $n_y = 2$. The methodology proposes eight linear models for initialization, thus $N = 8$. The number of hidden neurons is $n_{hidden} = 64$ for each MLP, but this is a tunable parameter as well as the number of hidden layers. As a design choice, we considered $\tanh$ activation functions for the first two layers, such that the weights vary smoothly. To prevent overfitting, the *NN WF* is trained and then validated every 100 epochs using the validation dataset.

E. Performance comparison with other modeling techniques

Fig. 5 compares performances of a single linear model, a polytopic model with a double sigmoid weighting function (13) and *PM-net* over the testing dataset. The loss function (14) is computed for the different modeling techniques with the true response of the system over the testing dataset. The linear model scores 2.071, the *PM-S* scores 0.666 while the *PM-net* outperforms the two other modeling techniques and get a loss value of 0.155. Furthermore, *PM-net* only needs five linear models while *PM-S* uses eight.

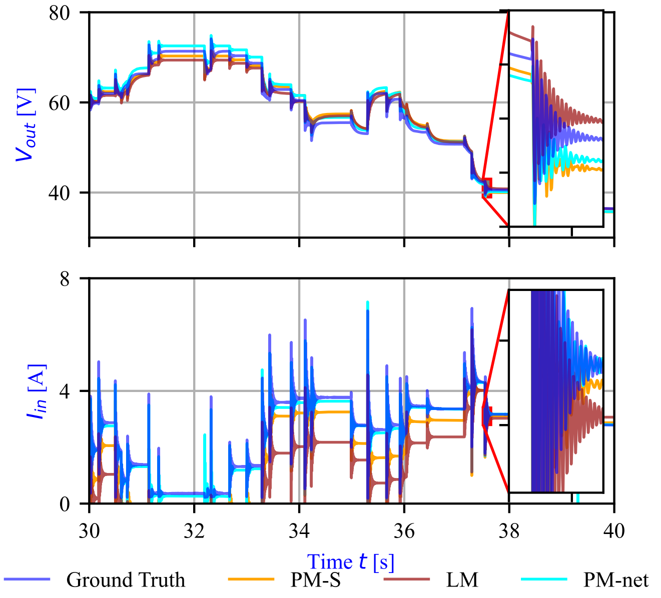

Fig. 5. Comparison between the different modeling techniques over 10 seconds out of 60 seconds of the testing dataset.

V. Conclusion

We have presented a new methodology for large-signal black-box modeling of power electronic converters. Based on polytopic models, we use neural networks to optimally combine linear models' responses to estimate the converter's dynamics over a large operating space. Although the methodology is demonstrated with a DC-DC converter, it can also be applied to DC-AC, AC-DC and AC-AC converters. The following conclusions can be drawn about the proposed methodology:

- It provides an effective way to re-partition the operating space to decrease model complexity at the price of minor performance degradation,
- It is able to suggest new operating points on which linear models can be identified to improve model's accuracy,
- The methodology is validated over a large dataset and the enhanced neural-network based polytopic model shows better results as compared to traditional methods.

Acknowledgements

Antonin Colot gratefully acknowledges the financial support of a research fellowship of the F.R.S.-FNRS.

References

[1] A. Francés, R. Asensi, Ó. García, 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.

[2] A. Francés, R. Asensi, Ó. García, and J. Uceda, "A blackbox large signal Lyapunov-based stability analysis method for power converter-based systems," in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*. IEEE, 2016, pp. 1–6.

[3] A. Francés, R. Asensi, Ó. 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.

[4] V. Valdivia, A. Lazaro, A. Barrado, P. Zumel, C. Fernandez, and M. Sanz, "Black-box modeling of three-phase voltage source inverters for system-level analysis," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3648–3662, 2011.

[5] A. Francés, R. Asensi, and J. Uceda, "Blackbox polytopic model with dynamic weighting functions for DC-DC converters," *IEEE Access*, vol. 7, pp. 160 263–160 273, 2019.

[6] 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.

[7] 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.

[8] P. Qashqai, K. Al-Haddad, and R. Zgheib, "Modeling power electronic converters using a method based on long-short term memory (LSTM) networks," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2020, pp. 4697–4702.

[9] 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.

[10] 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.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 116–132, 1985.

[12] M. Forgione and D. Piga, "*dynoNet*: A neural network architecture for learning dynamical systems," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 4, pp. 612–626, 2021.