Black-Box Modelling of a DC-DC Buck Converter Based on a Recurrent Neural Network

Gabriel Rojas-Dueñas
Universitat Politècnica de Catalunya
Terrassa, Spain
gabriel.esteban.rojas@upc.edu

Jordi-Roger Riba
Universitat Politècnica de Catalunya
Terrassa, Spain
riba@ee.upc.edu

Khaled Kahalerras
Airbus Operations S.A.S.
Toulouse, France
mohamed-khaled.kahalerras@airbus.com

Manuel Moreno-Eguilaz
Universitat Politècnica de Catalunya
Terrassa, Spain
manuel.moreno.eguilaz@upc.edu

Akash Kadechkar
Universitat Politècnica de Catalunya
Terrassa, Spain
akash.kadechkar@upc.edu

Alvaro Gomez-Pau
Universitat Politècnica de Catalunya
Terrassa, Spain
alvaro.gomez-pau@upc.edu

***Abstract*— Artificial neural networks allow the identification of black-box models. This paper proposes a method aimed at replicating the static and dynamic behavior of a DC-DC power converter based on a recurrent nonlinear autoregressive exogenous neural network. The method proposed in this work applies an algorithm that trains a neural network based on the inputs and outputs (currents and voltages) of a Buck converter. The approach is validated by means of simulated data of a realistic nonsynchronous Buck converter model programmed in Simulink and by means of experimental results. The predictions made by the neural network are compared to the actual outputs of the system, to determine the accuracy of the method, thus validating the proposed approach. Both simulation and experimental results show the feasibility and accuracy of the proposed black-box approach.**

Keywords—neural network, power converter, training, prediction, system identification, black-box model

I. Introduction

Nowadays, modelling and identification of power converters is a challenge, because most manufacturers do not provide the topology nor the parameters of the electrical components. This represents an issue for different industry sectors requiring to predict the dynamics and performance of the power systems they are using. The performance of the power converter can be estimated by identifying the parameters of the electrical components or by finding a system or architecture that emulates its behavior [1].

In most cases, the measurements at the input and output terminals of the power converter are the only sources of information. Therefore, when dealing with power converters, currents and voltages at the input and output terminals are the base for the different identification methods. The approaches that deal with the identification of the converter performance depend on the degree of accuracy sought and on the availability of the information. The power converters can be modelled as white-box, grey-box or black-box model depending on different aspects [2].

The white-box model assumes that the analytical and theoretical model are known, as well as the physical properties of the power converter. It uses the information and the measured currents and voltages in order to obtain the values of the electrical components that best describe the steady state and dynamic behavior of the circuit [3]. The second model is the grey-box. It is not completely described by physical equations, but equations and parameters are physically interpretable; it is a composition of white- and black-box models [4].

The black-box model is the one considered in this work. There is no physical knowledge about the system and the inputs are related to the outputs by only using experimental data. Different methods have been used to identify the performance of black-box models, their complexity depending on the linearity of the problem. Traditional methods such as ARMAX (AutoRegressive Moving-Average with eXogenous) and NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous input) can correctly identify the performance of the model for a known dataset using an autoregressive moving average algorithm that uses the past states of the variables to generate future outputs, which can be based on support vector machines (SVM) [5], [6]. However, when they are used to predict the values of the output variables based on new input data values, the accuracy tends to decrease and the error can be substantial. In [7] three different black-box modeling techniques for power converters are presented. The first one consists on building a G-Parameters model, by assuming that converters are classified as a two port network. Nevertheless, this approach is only valid for linear models, which is not the case of the buck converter analyzed in this paper. The second technique refers to the Wiener-Hammerstein model, which identifies two transfer functions that are able to reproduce the static nonlinearities and linear dynamics of the converter. The polytopic model is the last modeling technique, which identifies the stationary and transient behaviors of the converter. However, its complexity is significant if a great accuracy is required.

Considering the black-box as an artificial neural network, the identification process deals with the characteristics of nonlinear models, while allowing to predict effectively the model outputs [8]. The nonlinear autoregressive exogenous neural network (NARX NN) is a possible solution for this problem. In [9], this method is used to predict the daily direct solar radiation based on atmospheric conditions at a certain point. In addition, [10] uses NARX neural networks to forecast the wind speed in a certain area, having as inputs the wind direction, pressure, temperature and solar radiation. The approach proposed in this paper is based on applying an algorithm that identifies the characteristics and structure of a NARX NN, which emulates the behavior of a DC-DC Buck converter. Once the neural weights and connections of the network are obtained after the training stage, the response of the power converter for any set of inputs can be estimated with high accuracy.

The method presented in this paper is advantageous because of the robustness of the NN, which is able to correctly predict the outputs of the power converter for new operating points which are not included in the training set, thus differing

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

from past methods that can only generate a response based on the data used for identification. Furthermore, the model is easy to reproduce in simulations because it consists of a mathematical model that represent the weights of the neurons. This is useful because it allows the industrial users to simulate the behavior of its converters even when the manufacturer does not provide relevant information about the converters, which allows developing more accurate predictive maintenance plans, energy forecasting, etc.

II. Recurrent Nonlinear Autoregressive Exogenous Neural Network Overview

A black-box system is a structure with an unknown topology and/or parameters, which is often interpreted by means of the input and output signals [1]. In general, the outputs are the result of an excitation or stimulus applied to the black-box in the form of inputs values or vectors. It is not an easy problem to estimate the topology and parameters of a system where only the input and output values are known.

Recurrent neural networks (RNNs) work as a model that deals with the dynamics of a system. The main characteristic is that RNNs feedback the output variables to the input of the black box system. A Recurrent NARX NN (or NARX-RNN) arises as a solution for the identification of a black-box system performance. It can predict the output of a nonlinear system based on the expansion of past inputs and outputs. The model depends on the configuration of the system, and can be defined either by one of the two following equations:

$$\hat{y}(t) = f_N[y(t-1),...,y(t-n_a),u(t-1),...,u(t-n_k)] \quad (1)$$

$$\hat{y}(t) = f_N[\hat{y}(t-1),...,\hat{y}(t-n_a),u(t-1),...,u(t-n_k)] \quad (2)$$

where $y(t)$ and $\hat{y}(t)$ refer to the black-box actual and simulated outputs (autoregressive) respectively, $u(t)$ is the system input (exogenous variable), n_a refers to the number of output delays and n_k the number of input delays [9].

The two architectures of the NARX-RNN system, defined by equations (1) and (2), depend on whether or not the output feeds-back the actual output information. The first one is the open loop (or series) architecture, which is mostly used for design and training the neural network, because it uses *a priori* target information (known) to estimate the output of the system (predicted). On the other hand, the closed loop architecture uses the actual estimated output as an input of the black-box, so there is no need to have the target data beforehand, which is useful for predictions [9]. Fig. 1 shows both architectures.

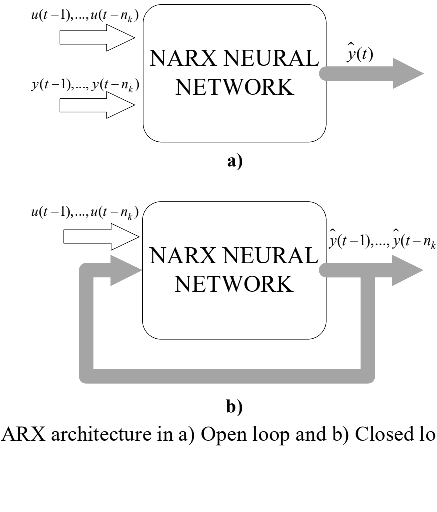


Fig. 1. NARX architecture in a) Open loop and b) Closed loop.

The structure of a neural network consists of one input layer, an output layer and the hidden layers, whose number is difficult to determine, and depends on the type of problem, thus it may change depending on the application and the nature of the input data.

Fig. 2 presents the NARX-RNN topology with an open loop architecture. The complexity of the problem relies on the number of inputs, outputs, delays and neurons on each layer, since these parameters define the total number of variables (connections) of the optimization problem to be solved.

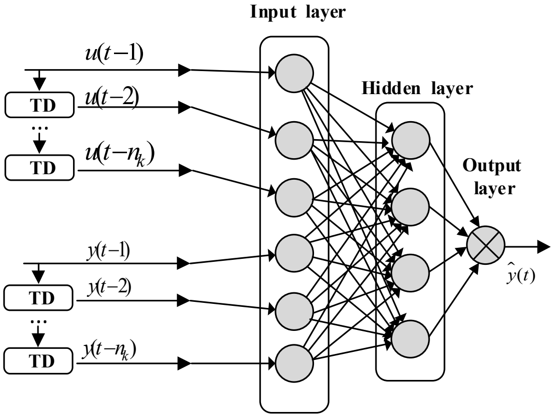


Fig. 2. NARX Neural Network structure.

An iterative process is carried out to calculate the neuron weights, where in every epoch the values of the weights are updated depending on the root mean squared error obtained. The Levenberg-Marquardt algorithm is used for this purpose. The main drawback is that it calculates the Jacobian and the Hessian at each iteration. Then, for large data sets and networks, the matrices can be significantly large, thus requiring a considerable amount of computing resources.

A. Model structure

The input and output variables of the black-box system were chosen considering that the trained NARX-RNN emulates the behavior of a DC-DC power converter. Then, if the topology of the converter is unknown, the only available data are the current and voltage measurements at the input and at the output terminals of the circuit. However, the values of the input and output variables of the converter do not necessarily match with the ones of the NARX neural network.

Considering that the variables can be modified externally, the black-box signals for this particular problem are classified as follows,

- Inputs: input voltage and output current, which can be controlled externally by means of the voltage source and the load, respectively.
- Outputs: input current and output voltage of the power converter, which depend on the input variables.

III. Algorithm to Train the NARX-RNN

It is difficult to know *a priori* the exact configuration of a NARX-RNN that fits the black-box dynamic characteristics, because it depends on different parameters of the dataset. Thus, this section proposes an algorithm in order to obtain a NARX-RNN that is able to predict the values of the output variables based on the input variables of the model. Fig. 3 shows the flowchart of this algorithm.

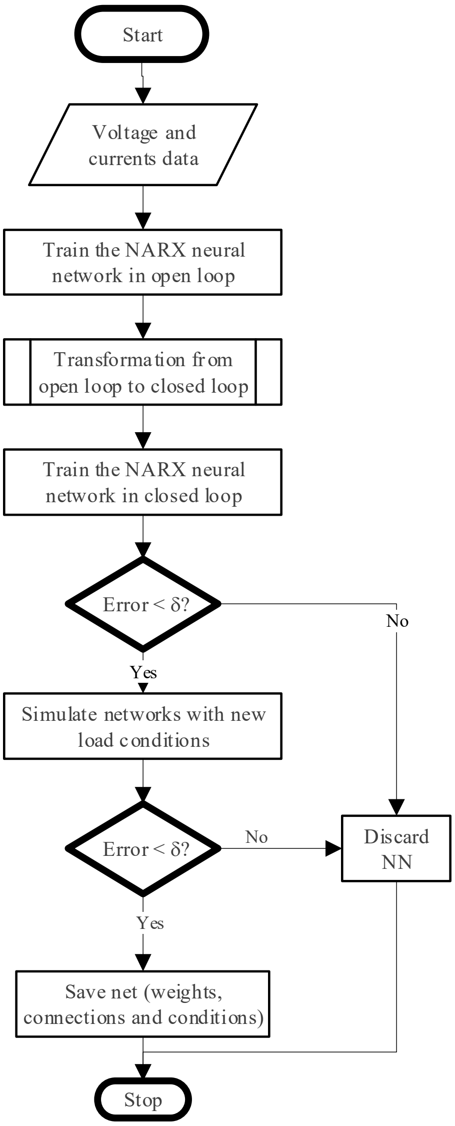

Start
Voltage and currents data
Train the NARX neural network in open loop
Transformation from open loop to closed loop
Train the NARX neural network in closed loop
Error < δ?
No
Yes
Simulate networks with new load conditions
Error < δ?
No
Discard NN
Yes
Save net (weights, connections and conditions)
Stop

Fig. 3. Algorithm to train the NARX-RNN.

It is important to state the specifications of the measured data because they determine the accuracy and effectiveness of the training algorithm. The data set must consider different load scenarios in order to cover all the space of possible outputs. In addition, the time step must be the same for all sets of values. Interpolation is required when the time step is not constant.

Many neural networks were tested by varying their parameters, and the one with the lowest error was chosen. The parameters and their importance in the neural network are described as follows,

- Number of delays: this is the parameter that defines the number of previous time steps to be considered in order to calculate the present value of the output variables. It directly affects the time required to train the network since it adds inputs to the black-box.
- Number of neurons in the hidden layer: it results difficult to determine the exact number of neurons of the hidden layer, leading to a trial and error process. It greatly depends on the architecture and parity of the network [11]. Furthermore, the computational time required to train the model is lower with neural networks with a small number of neurons, which is desirable for this work.
- Length of the dataset: it is one of the most important parameters because too low or too large number of points may lead to underfitting or overfitting, respectively. It is noted that each point represents a time step of the signal. The dataset consists on n number of experiments, where each experiment presents a load change.
- Training ratio: this parameter gives the percentage of the dataset that is designated for training, validation and testing. If the training ratio is x%, the validation and test percentage corresponds to $(100 - x)/2$ %. This value tends to change with the length of the dataset.

IV. Simulations

The DC-DC Buck converter was chosen to generate a training dataset that represents the static and dynamic behavior of the model.

Fig. 4 presents the topology of the converter. It includes some parasitic elements such as the output impedance of the voltage source (L_{in} and R_{in}) [12], the equivalent series resistance (ESR) of the capacitors (R_{Cin}, R_{C1} and R_{C2}) and the series resistance of the inductor (R_L).

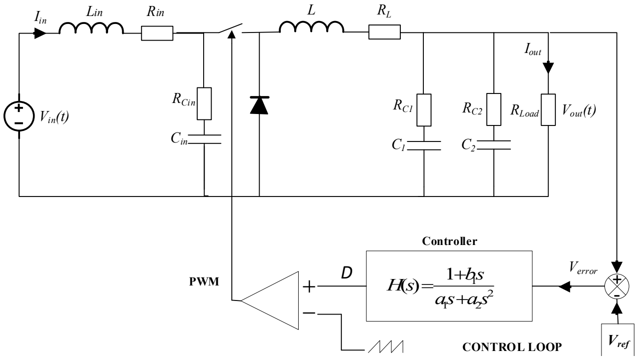


Fig. 4. Buck converter topology.

All the parameters of the power converters are fixed, besides the load. Table I shows the values of each element of the circuit, which can be found in the datasheet of the TPS40200EVM-002 buck converter (Texas Instruments).

TABLE I. Buck Converter Parameters

Parameter	Value
L	33 µH
$C1$	20 µF
$C2$	440 µF
R_L	60 mΩ
R_{c1}	65 mΩ
R_{c2}	300 mΩ
R_s	40 mΩ
C_{in}	28.47 µF
L_{in}	0.3018 µH
R_{cc}	5273.7 Ω
R_{cin}	100 mΩ
R_{in}	434.9 mΩ
V_{out}	3.3 V
Frequency	197 kHz

To test the performance of the proposed algorithm, the power converter was modelled and simulated in MATLAB/Simulink using the Simscape Electrical library. The input voltage was fixed at 19 V, and the load was controlled externally by means of a variable resistor, which allows generating different load conditions. The simulation time was 0.2 seconds with multiple variations of the load at different frequencies to provide randomness to the training data and avoiding an incorrect identification of the neural network parameters. The simulations were performed in continuous time and then the data was interpolated to have a dataset with the same fixed step for all the experiments.

The response of the system was obtained, as shown in Fig. 5 by applying a load change, which includes both, steady state and transient responses. This is just one case of the 500 that were simulated for the training of the neural network. It is important to mention that the output load range was selected to operate the Buck converter in continuous conduction mode (CCM). To guarantee that all the simulations were performed in the CCM of the converter, a parametric sweep of the load value was performed while measuring the current flowing across the inductor. Then, the maximum value of the output resistor was fixed to 15.8 Ω, which is the value before the inductor current is almost null.

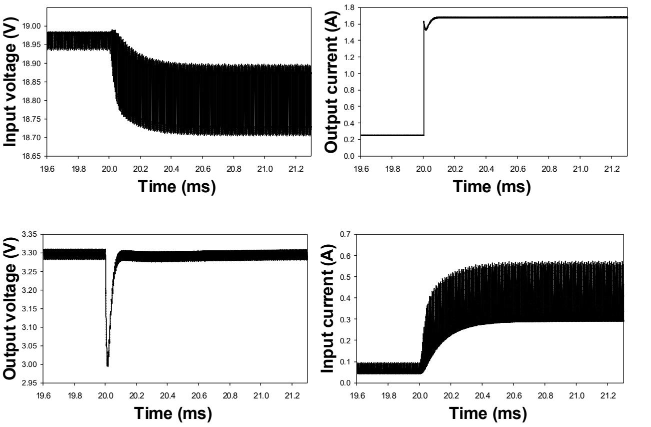


Fig. 5. A) Input voltage, b) Output current, c) Output voltage and d) Input current of the Buck converter.

After obtaining a rich dataset of the input and output variables of the black-box model, the next step is to specify the fixed and changing parameters of the neural networks to be trained. For this particular case, the delays are fixed and the parameters that vary are the number of neurons, length of the dataset and the percentage assigned to training, validation and test. Table II shows the aforementioned parameters.

TABLE II. NEURAL NETWORK CONFIGURATION

Parameter	Value
Maximum number of epochs	1000
Maximum time	3 hours
Minimum error	1e-7
Maximum number of validation failures	6
Neurons in hidden layer	5, 6, 7, 8, 9, 10, 14, 20, 30
Length (Number of experiments)	100, 120, 140, 150, 160, 180,200, 210, 220, 230, 240, 250
Training ratio (percentage)	40, 45, 50, 60, 65,70, 75, 80. 85, 90, 100

A total of 176 NARX-RNN were trained, requiring four days of computation, using an Intel Xeon CPU E5-1650 v2 @3.5 GHz, which means an average simulation time of 45 minutes per network. In open loop configuration, every network managed to obtain an error value below 0.0001, calculated according to root mean square error (RMSE). However, after the transformation to closed loop and the new training stage, the error increased and just seven of them (3.98% of the proposed neural networks) had an RMSE value below 0.001.

Considering the neural networks with the lowest error, there is not a clear relationship between their training parameters. The only clear conclusion that may be derived from the training of the 176 networks is that the ones with more neurons (more than 10) failed to reproduce correctly the target values of the variables, thus providing a relatively high error. In addition, the error value of the neural networks trained with more than 230 points and less than 140 points was higher due to underfitting and overfitting. On the other hand, there is not a clear pattern regarding the training ratio, mainly because it is strongly related to the length of the dataset that is used for training.

To determine the neural network that best approximates the performance of the Buck converter, the networks were tested under a new set of load values. Ten new load profiles, as the one shown in Fig. 5, were simulated. For the ten cases, the outputs of the neural networks and the estimated outputs were compared to the ones obtained from the Simulink simulations. The network that best fitted the ten scenarios consist of 10 neurons in the hidden layer, was trained with 200 experiments and a training ratio of 50%, thus obtaining a RMSE of $2.15 \cdot 10^{-4}$. The structure is shown in Fig. 6.

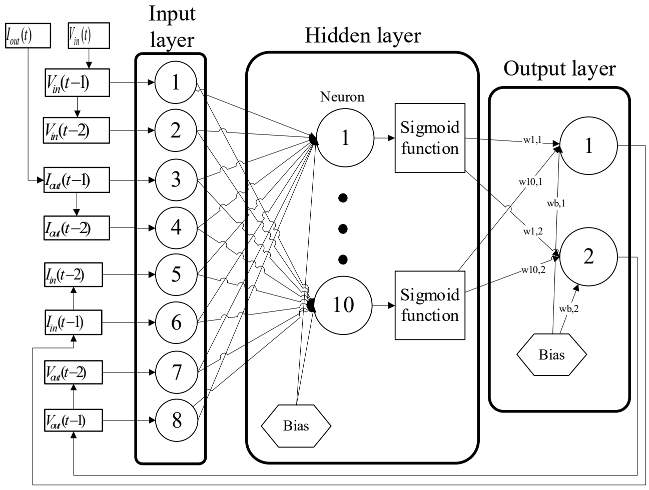
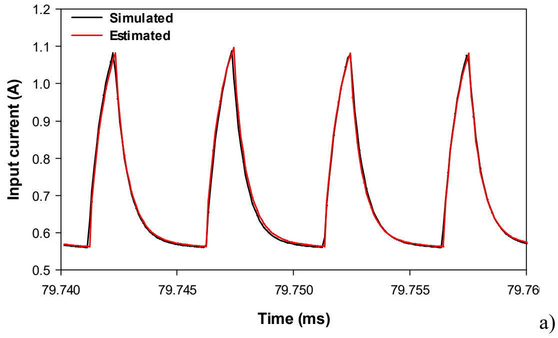


Fig. 6. Configuration of the resulting neural network.

Since the ten tests have multiple load conditions and 2 million of points each, it is necessary to focus on the steady state and a load change in order to compare the actual and estimated output values. Fig. 7 shows the estimated and simulated outputs when the load is set to 1.204 Ω (steady state) and Fig. 8 shows the comparison when there is a load change from 1.204 Ω to 0.791 Ω (transient state).



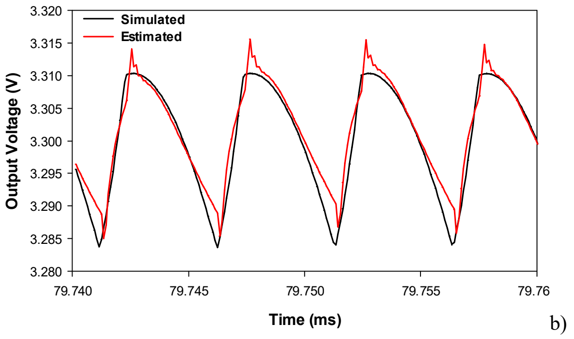


Fig. 7. Actual and estimated response. a) Input current in steady state. b) Output voltage in steady state.

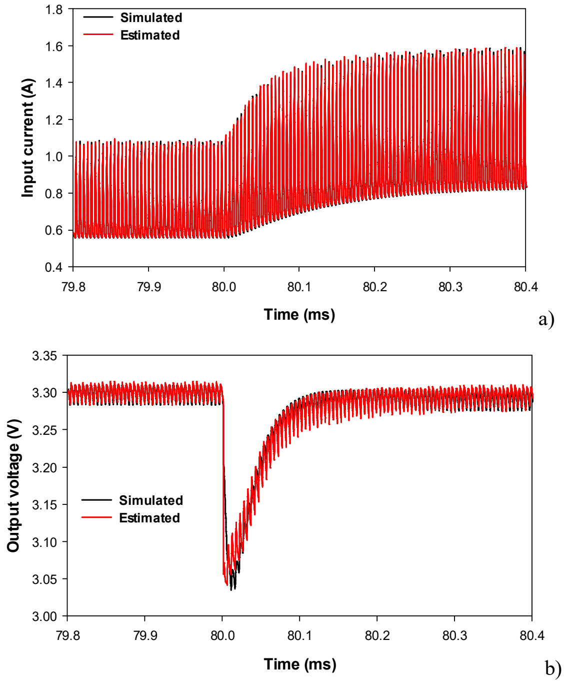


Fig. 8. Actual and estimated response. a) Input current during a load change. b) Output voltage during a load change.

The results presented in Fig. 7 and Fig. 8 show that there is almost no difference between the predicted input current of the neural network and the one simulated in Simulink. There is no difference that can be appreciated and even with a change in the load, the system predicts the transient state with accuracy. On the other hand, the output voltage estimation is not as accurate as the one of the current. However, the mean value and the ripple are exactly the same, the only slight difference under steady state conditions being the shape of the waveform. When the load change occurs, the neural network system predicts the overshoot but the response time is marginally faster.

V. Experimental Results

The next step is to test the proposed algorithm using experimental data. For this purpose, the Texas Instruments TPS40200EVM - 002 nonsynchronous Buck converter was used. Fig. 4 and Table I present the topology and the values of the passive components provided by the manufacturer, respectively.

Considering that a rich dataset is needed to train the neural networks, a large number of experiments is required. It is noted that each experiment represents a load change at the converter output. This leads to the implementation of an automatic system that controls the value of the output load. The load consists of eight resistors connected in parallel, each one of them can be connected or disconnected by means of an Arduino microcontroller board, resulting in 256 different load conditions. In addition, the oscilloscope is also controlled externally and the data of each experiment is saved automatically in a computer. Fig. 9 shows the experimental setup.

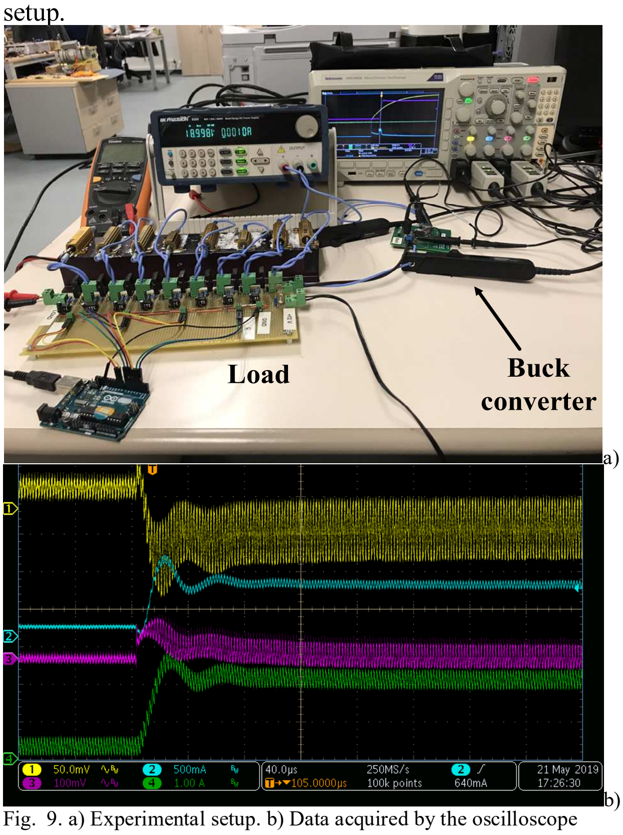


Fig. 9. a) Experimental setup. b) Data acquired by the oscilloscope

Measurements were performed by means of a DC power supply (BK Precision 9205; BK Precision Corporation, Yorba Linda, CA, USA). Input/output voltages and currents were acquired by using of a four channel oscilloscope (Tektronix MDO3024 200 MHz 2.5 GS/s; Tektronix, Beaverton, OR, USA), two high-frequency current probes (Tektronix TCP0030A 0.001-20 A 120 MHz; Tektronix, Beaverton, OR, USA) and two high-frequency voltage probes (Tektronix TPP0250 250 MHz; Tektronix, Beaverton, OR, USA).

The values of the resistors used in the measurements were 30 Ω (two units), 25 Ω (two units), 18.7 Ω (two units), 8 Ω and 4 Ω. The Arduino was programmed to randomly generate load conditions in the range of the continuous conduction mode of the converter, which is between 0 Ω and 15 Ω. A total of 500 experiments were taken, half of them for the training process of the NARX networks and the other half to test the networks with the lowest error in order to select the best one.

Before training the nets as described in Section IV but this time by using the experimental data, a preprocessing stage is necessary to filter and shorten the data. The data of each experiment is filtered using a resizing and a low- pass filter, resulting in faster simulations because the number of points per experiment is reduced. This step results fundamental for the correct identification of the NARX network because the noise present in the experimental data may affect the training of the NNs. This occurs because the NN starts to learn from the noise instead of the relevant information (overfitting).

To train the NARX networks using the experimental data, the same procedure as the one depicted in Sections III and IV is carried out. The best networks were tested with a new dataset and the one that showed the lowest RMSE was trained with 200 experiments and a training ratio of 70%, it has 8 neurons in the hidden layer and a RMSE of $2.15 \cdot 10^{-4}$. The resulting network follows the same structure as the one in Fig. 6 with the only difference that there are eight neurons in the hidden layer. Fig. 10, compares the measured and estimated curves for one experimental load change.

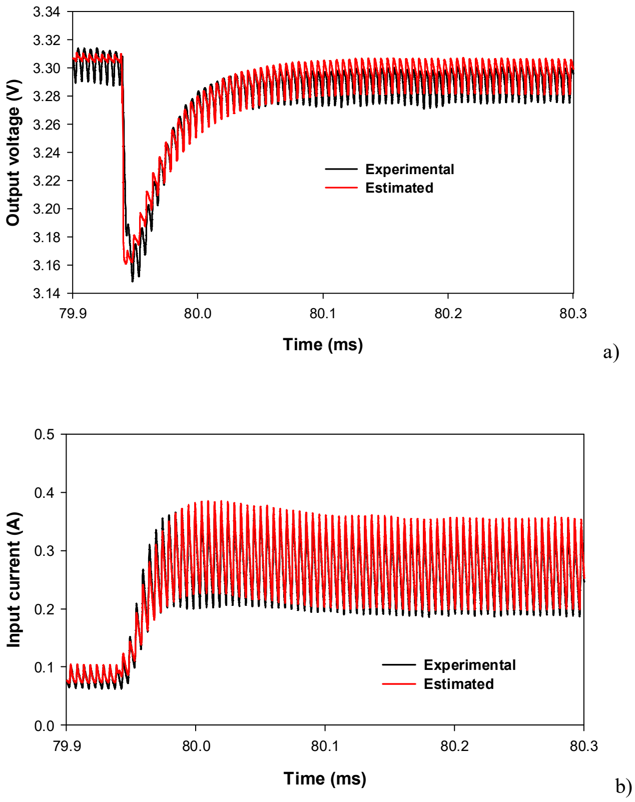


Fig. 10. Measured and estimated a). Output voltage. b) Input current.

Figs. 10 show a significant resemblance between the measured and the curves predicted by the black-box. This is just one of the 250 experiments where the net was tested after the training. To have a better understanding of the trained NARX net accuracy, the coefficient of determination R^2 between measured and estimated data was calculated for the 250 test experiments. Fig. 11 and Table III show the results.

TABLE III. COEFFICIENT OF DETERMINATION R^2 FOR THE 250 TEST EXPERIMENTS

Variable	$\overline{R^2}$	$min(R^2)$	$max(R^2)$
I_{in}	0.9876	0.9464	0.9967
V_{out}	0.9815	0.9322	0.9938

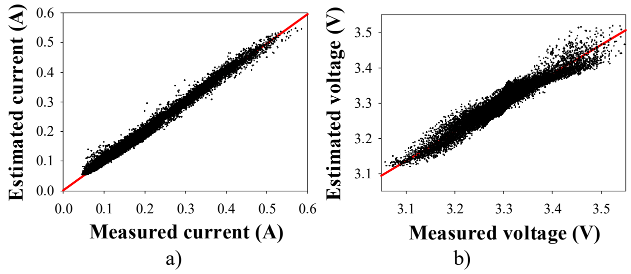


Fig. 11. Scatter plot between measured and estimated data. a) Input current; b) Output voltage.

From the results presented above it is evident that there exists a good correlation (R^2 is close to 1) between the measured and estimated data when the dataset is tested with new operating points.

VI. CONCLUSIONS

This paper has presented the identification of a recurrent NARX neural network that emulates the behavior of a DC-DC Buck power converter with an unknown topology. This estimation has been done by using a simulated dataset under different load conditions across a window of time. Since there is no an exact method to determine the configuration of the best neural network, different networks where tested and the one with the lowest error was selected. The final result is the weights of the connections between the different layers of the system and a trained network that is able to predict the output of the power converter for a given input voltage and load. Experimental data has also proved the accuracy of the proposed black-box model.

The algorithm proposed in this work will be further extended to new types of converters and experimental data.

ACKNOWLEDGMENT

This work has been partially supported by the Generalitat de Catalunya under the project 2017SGR0967, the Spanish Ministry of Economy and Competitiveness under the project TRA2016-80472-R and the European Commission through the Clean Sky program, under the project 755332-AEMS-IdFit.

FUNDING

This project has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation program under grant agreement No 755332.

Clean Sky 2 JOINT UNDERTAKING

This project was funded by the European Union

DISCLAIMER

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Reproduction is authorized provided the source is acknowledged.

REFERENCES

[1] V. Valdivia, A. Barrado, A. Laazaro, P. Zumel, C. Raga, and C. Fernandez, "Simple Modeling and Identification Procedures for 'Black-Box' Behavioral Modeling of Power Converters Based on Transient Response Analysis," *IEEE Trans. Power Electron.*, vol. 24, no. 12, pp. 2776–2790, Dec. 2009.

[2] H. Balakrishnan, M. Moreno-Ezuilaz, J.-R. Riba, S. Bogarra, and A. Garcia, "DC-DC Buck Converter Parameter Identification Based on a White-Box Approach," in *2018 IEEE 18th International Power Electronics and Motion Control Conference (PEMC)*, 2018, pp. 242–247.

[3] J. Riba, M. Moreno-eguilaz, S. Bogarra, and A. Garcia, "Parameter

Identification of DC-DC Converters under Steady-State and Transient Conditions Based on White-Box Models," *Electronics*, vol. 7, no. 393, 2018.

[4] R. Kicsiny, "Grey-box model for pipe temperature based on linear regression," *Int. J. Heat Mass Transf.*, vol. 107, pp. 13–20, Apr. 2017.

[5] A. Rahrooh and S. Shepard, "Identification of nonlinear systems using NARMAX model," *Nonlinear Anal. Theory, Methods Appl.*, vol. 71, no. 12, pp. e1198–e1202, Dec. 2009.

[6] G. Acuna, C. Ramirez, and M. Curilem, "Comparing NARX and NARMAX models using ANN and SVM for cash demand forecasting for ATM," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–6.

[7] A. Frances, R. Asensi, O. Garcia, R. Prieto, and J. Uceda, "Modeling electronic power converters in smart dc microgrids - An overview," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6274–6287, Nov. 2018.

[8] Yang-Yin Lin, Jyh-Yeong Chang, and Chin-Teng Lin, "Identification and Prediction of Dynamic Systems Using an Interactively Recurrent Self-Evolving Fuzzy Neural Network," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 2, pp. 310–321, Feb. 2013.

[9] Z. Boussaada, O. Curea, A. Remaci, H. Camblong, and N. M. Bellaaj, "A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation," *Energies*, vol. 11, no. 3, 2018.

[10] E. Cadenas *et al.*, "Wind Speed Prediction Using a Univariate ARIMA Model and a Multivariate NARX Model," *Energies*, vol. 9, no. 2, p. 109, Feb. 2016.

[11] D. Hunter *et al.*, "Selection of Proper Neural Network Sizes and Architectures — A Comparative Study," *IEEE Trans. Ind. Informatics*, vol. 8, no. 2, pp. 228–240, 2012.

[12] A. Prakash, "Benefits of a Low Inductive Shunt for Current Sensing in PWM Applications Benefits of a Low Inductive Shunt for Current Sensing in PWM Applications," 2017.