Computationally Efficient Sphere Decoding Algorithm based on Artificial Neural Networks for Long-Horizon FCS-MPC

Eduardo Zafra, *Student Member, IEEE*, Joaquin Granado, Vicente Baena Lecuyer, Sergio Vazquez, *Fellow, IEEE*, Abraham M. Alcaide, *Member, IEEE*, Jose I. Leon, *Fellow, IEEE* and Leopoldo G. Franquelo, *Life Fellow, IEEE*

Abstract

Successful application of finite control set model predictive control (FCS-MPC) strategies with long prediction horizon depends on the careful design of the optimization algorithm. The conventional method involves transforming the problem to an equivalent box-constrained integer least-squares (BILS) formulation that can be solved with branch-and-bound techniques such as the sphere decoding algorithm (SDA). In this work, it is proposed to define an artificial neural network (ANN) to replace the SDA, avoiding its inherent computational variability. Similarly to practical applications of the SDA, the ANN finds an approximate solution of the underlying optimization problem. In contrast, the main benefit of the proposed approach is that it can be implemented in a low-cost microprocessing platform, greatly improving the performance in terms of resources in comparison with other advanced techniques proposed in the literature.

Index Terms

Neural network applications, predictive control, DC-AC power conversion.

This is the accepted version of the paper: E. Zafra et al., "Computationally Efficient Sphere Decoding Algorithm Based on Artificial Neural Networks for Long-Horizon FCS-MPC," in IEEE Transactions on Industrial Electronics, vol. 71, no. 1, pp. 39-48, Jan. 2024, doi: 10.1109/TIE.2023.3243301.

I. Introduction

MODEL predictive control (MPC) is becoming a prominent research topic in the power electronics field [1], [2]. Particularly, finite control set - MPC (FCS-MPC) strategies are favored in the literature thanks to its straightforward formulation and ease to address the control of complex systems and multiple control targets [3].

In a power converter, there is a set of admissible switching states defined by the topology. This is the so-called finite control set (FCS). Considering control inputs constrained to the FCS, allows one to relatively reduce the problem formulation complexity in comparison to continuous control set MPC methods [4]. Conveniently, in FCS-MPC, the problem is reduced to an exhaustive search algorithm (ESA), i.e. evaluating the FCS candidates in terms of system plant predictions and a cost function that determines the suitability of each candidate. However, exhaustive enumeration of all the possible candidates can become intractable depending on the selected prediction horizon length (N_p), i.e. the number of time steps considered in these predictions. In particular, the problem complexity grows exponentially with N_p. Thus, an ESA-based solution is generally disregarded due to computational issues for $N_p \geq 3$. There is an important amount of recent research works attempting to mitigate this problem [2], [5]–[7]. The reason is that important closed-loop performance benefits can be achieved by increasing the prediction horizon. In particular, it is possible to achieve a better harmonic performance for a given switching frequency, or reduce the switching frequency for a target harmonic distortion [8], [9].

The authors gratefully acknowledge the financial support provided by the project PID2020-115561RB-C31 funded by MCIN/AEI/10.13039/501100011033 and the project project TED2021-130613B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR". Eduardo Zafra gratefully acknowledges the financial support provided by the Spanish Ministry of Universities under grant FPU18/02704. Abraham M. Alcaide gratefully acknowledges to the Contratación de Personal Investigador Doctor. (Convocatoria 2019) 43 Contratos Capital Humano Línea 2. Paidi 2020, supported by the European Social Fund and Junta de Andalucía.

E. Zafra, J. Granado, V. B. Lecuyer and A. M. Alcaide are with the Electronics Department, Universidad de Sevilla, 41092 Sevilla, Spain (e-mail: ezafra1@us.es; j_granado@us.es; v_baena@us.es; amarquez@ieee.org).

S. Vazquez, J.I. Leon and L. G. Franquelo are with the Laboratory of Engineering for Energy and Environmental Sustainability, Universidad de Sevilla, 41092 Sevilla, Spain (e-mail: sergi@us.es; jileon@us.es; lgfranquelo@us.es).

To overcome these computational issues, a large part of the study of long prediction horizon -FCS-MPC (LPH-FCS-MPC) in the literature has focused on the design and implementation of the optimization algorithm. To solve this problem more efficiently than through exhaustive enumeration, it was proposed to transform the problem into an equivalent box-constrained integer least-squares (BILS) problem [8]. This enabled the usage of branch-and-bound techniques such as the sphere decoding algorithm (SDA) that can solve the problem without exploring every candidate in the FCS. Research works achieving practical implementation of LPH-FCS-MPC can be categorized in three groups. The first one is characterized by the use of dSPACE platforms, generally reaching up to N_p = 4 [5], [10], [11]. These platforms offer excellent rapid prototyping capabilities. However, they are quite expensive (around tens of thousands of dollars). The second group is based on low cost FPGA platforms (hundreds of dollars) [6], [12]–[14]. These implementations achieve great performance as they can generally reach N_p ranging from 5 to 7. Nonetheless, these proposals usually require hardware programming which is not ideal in terms of design and verification complexity. Finally, other works opt to use intermediate solutions following High-Level Synthesis tools to accomplish implementations in FPGA-based platforms with reduced design complexity [7], [15]. However, the required platforms in this case are more expensive as FPGA resource consumption is higher. Also, the obtained performance is generally lower, with N_p = 4.

In this paper, it is proposed to imitate the behavior of the SDA by means of an artificial neural network (ANN). The main objective is to establish a method by which LPH-FCS-MPC can be solved reaching large N_p with reduced costs and low computational requirements. ANN-based techniques have been previously proposed in FCS-MPC works. In [16]–[18], an ANN is used to select the optimal weighting factors of the cost function. However, the control algorithm is solved through conventional methods. It is possible to find some works in the literature that propose the usage of neural networks to imitate the control algorithm. However, there are some important differences that can be highlighted. In [19], an ANN-based control strategy is proposed for the output voltage control of an UPS. The ANN is trained with a conventional FCS-MPC strategy with N_p = 1. The design is validated with simulation results. Thus, there is no proof of experimental validity of this method or extension to long prediction horizons. This limits the proposal practical usefulness as FCS-MPC with $N_p = 1$ is not computationally challenging with modern control platforms. In [20], an ANN is proposed to learn an LPH-FCS-MPC problem up to $N_p = 5$. However, experimental assessment is not provided. This step is crucial as the main motivation for the usage of ANN is to obtain a more easily implementable algorithm to solve the LPH-FCS-MPC problem. In [21], an ANN is proposed to imitate the behavior of conventional FCS-MPC up to $N_p = 3$. Experimental validation is provided to show the validity of ANN for practical application. However, usage of the conventional FCS-MPC formulation limits the achievable prediction horizon. Also, control effort limitation is not considered in the formulation. This is an important element as it is known that under no control effort penalty, long horizon FCS-MPC renders the same solution as FCS-MPC with $N_p = 1$ [1].

In contrast to other FCS-MPC ANN imitation works, the proposed approach is trained to learn only the search stage of the LPH-FCS-MPC algorithm. This aims to be a more flexible and generalizable method. Particularly, a conventional SDA is selected as it provides optimality guarantee with much greater efficiency than exhaustive enumeration, while being a sufficiently standard solution in the LPH-FCS-MPC literature. Another benefit is that SDA imitation is relatively well-known in multiple input multiple output (MIMO) communication systems [22] where it is usually employed as information detector. However, SDA has become computationally prohibitive in modern MIMO communication systems and several works have been published on this topic proving that the application of deep learning to MIMO decoding can drastically reduce the complexity of the decoder while preserving its performance [23]–[26]. In particular, the authors of [23] investigate several ANNarchitectures, obtaining promising accuracy with low computational complexity. The major problem of the proposed architectures is their lack of flexibility, causing a performance drop of the detector in varying channels, a common situation in real communication systems but not in LPH-FCS-MPC for power conversion applications where the system model can usually be considered as time invariant [7]–[10].

Thanks to the proposed method, it is possible to achieve viable solutions up to $N_p = \{7, 8\}$ using a single microprocessing core of a low-cost platform such as the Zynq-7000. In contrast to other LPH-FCS-MPC works, this eliminates the requirement of costly platforms or hardware design knowledge, providing a very accesible methodology to implement LPH-FCS-MPC and incentivize its usage in the industry.

The rest of this paper is organized as follows: In Section II, the selected system model and the control problem are presented. Section III is dedicated to illustrate the conventional SDA algorithm. The ANN imitator for SDA solution is described in Section IV. Finally, experimental validation is provided in Section V.

II. System Model and LPH-FCS-MPC Formulation

A grid-forming two-level three-phase voltage source inverter (VSI) with output LC filter is selected as the case study for this paper. The electric circuit is shown in Fig. 1. System variables and parameters are summarized in Table I.

The prediction model for this system can be expressed in state-space form as shown in Table II [1]. By default, magnitudes are referred to the stationary $\alpha\beta$-frame for simplicity. A steady-state Kalman filter is used to estimate the output current value

[1] $\boldsymbol{I}_n$ and $\boldsymbol{O}_n$ are the identity and zero matrices of dimension $n \times n$, respectively. The matrix $\boldsymbol{J}$ is: $\boldsymbol{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Matrix $\boldsymbol{T}_{abc}^{\alpha\beta}$ is the well-known power invariant Clarke's transformation matrix.

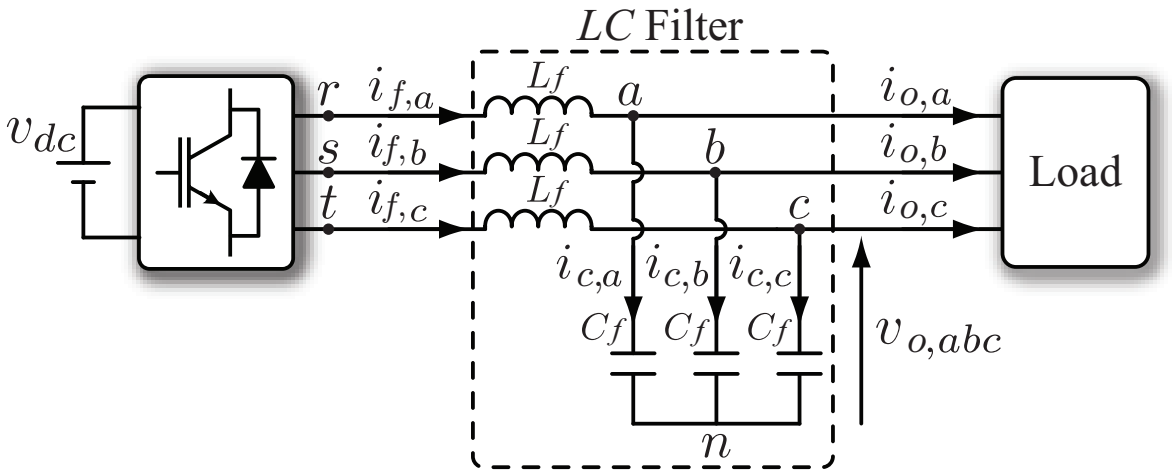


Fig. 1. Three-phase VSI with output LC filter circuit.

TABLE I
SYSTEM VARIABLES AND PARAMETERS

Variable	Description
L_f, C_f	Output filter inductor and capacitor
$\boldsymbol{i_f}_{,abc} = [i_{f,a}\ i_{f,b}\ i_{f,c}]^T$	Output filter inductor current
$\boldsymbol{v_o}_{,abc} = [v_{o,a}\ v_{o,b}\ v_{o,c}]^T$	Output filter capacitor voltage
$\boldsymbol{v_i}_{,abc} = [v_{i,a}\ v_{i,b}\ v_{i,c}]^T$	VSI output voltage
v_{dc}	DC-link voltage
$\boldsymbol{S}_{abc} = [S_a\ S_b\ S_c]^T$	VSI switching state
$\boldsymbol{i_o}_{,abc} = [i_{o,a}\ i_{o,b}\ i_{o,c}]^T$	Output load current
$\boldsymbol{v}^*_{\boldsymbol{o},abc} = [v^\star_{o,a}\ v^\star_{o,b}\ v^\star_{o,c}]^T$	Output filter capacitor reference voltage
f_s, T_s	Sampling frequency and sampling period

TABLE II
DISCRETE PREDICTION MODEL

	$\alpha\beta$ frame UPS system equations
State-space model	$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k$ $\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k$
State variables	$\boldsymbol{x}_k = [(\boldsymbol{i_f}_{,k})^T\ (\boldsymbol{v_o}_{,k})^T\ (\boldsymbol{i_o}_{,k})^T]^T$
Input	$\boldsymbol{u}_k = \boldsymbol{S}_{abc,k}$
Output and reference	$\boldsymbol{y}_k = \boldsymbol{v_o}_{,k}\ ,\ \ \boldsymbol{y}^\star_k = \boldsymbol{v}^\star_{\boldsymbol{o},k}$
Matrices	$\boldsymbol{A} = e^{\boldsymbol{A}_c T_s};\ \boldsymbol{B} = \int_0^{T_s} e^{\boldsymbol{A}_c(T_s-\tau)}\boldsymbol{B}_c d\tau$ $\boldsymbol{A}_c = \begin{bmatrix} \boldsymbol{O}_2 & -\frac{1}{L_f}\boldsymbol{I}_2 & \boldsymbol{O}_2 \\ \frac{1}{C_f}\boldsymbol{I}_2 & \boldsymbol{O}_2 & -\frac{1}{C_f}\boldsymbol{I}_2 \\ \boldsymbol{O}_2 & \boldsymbol{O}_2 & \boldsymbol{J}\omega \end{bmatrix}$ $\boldsymbol{B}_c = \begin{bmatrix} \frac{1}{L_f}\boldsymbol{I}_2 \\ \boldsymbol{O}_2 \\ \boldsymbol{O}_2 \end{bmatrix} \boldsymbol{T}^{\alpha\beta}_{abc} v_{dc},\ \boldsymbol{C} = \begin{bmatrix} \boldsymbol{O}_2 \\ \boldsymbol{I}_2 \\ \boldsymbol{O}_2 \end{bmatrix}^T$

without direct measurement [27]. The filter inductor current, capacitor output voltage and the output load current are the state variables.

The control input $\boldsymbol{u}_k$ is the switching state of the inverter, $\boldsymbol{S}_{abc,k}$ with $\boldsymbol{S}_x|_{x=a,b,c} \in \{0,1\}$. Thus, $\boldsymbol{u}_k \in \mathbb{V}$, where $\mathbb{V} = \{0,1\}^3$ is the FCS which defines the admissible switching positions. The system output $\boldsymbol{y}_k$ is a selection of the state variables that the controller aims to regulate. In the particular case of the grid-forming VSI, the output voltage $\boldsymbol{v_o}_{,k}$ is selected as the system output. The objective is to follow a sinusoidal reference $\boldsymbol{v}^\star_{\boldsymbol{o},k}$. For the LPH-FCS-MPC formulation, it is useful to define the so-called switching or input sequence $\boldsymbol{U}_k$, output sequence $\boldsymbol{Y}_k$ and output reference sequence $\boldsymbol{Y}^*_k$. These result from appending

the individual inputs and outputs within the prediction horizon in a single array [6], [12].

The desired behavior is imposed by means of a cost function. This function assigns a value to each possible control action that quantifies its suitability and impact in the system to accomplish the control targets. The same conventional cost function defined in [8] is used:

$$J_k = \sum_{\ell=k}^{k+N_p-1} \|\boldsymbol{y}_{\ell+1} - \boldsymbol{y}^{\star}_{\ell+1}\|_2^2 + \lambda\|\boldsymbol{u}_{\ell} - \boldsymbol{u}_{\ell-1}\|_2^2. \tag{1}$$

The output voltage regulation is addressed by the first term, as the squared error between output and reference is evaluated. Switching sequences that yield smaller tracking error will present lower costs and be favored by the optimization algorithm. The second term is the control effort penalization. It evaluates the switching effort of imposing a different switching state. Weighting factor $\lambda \geq 0$ adjusts the trade-off between the switching effort penalization and output tracking error. Higher values provide lower average switching frequency $\bar{F}_{sw}$ as states that imply switching a semiconductor are deemed less preferable. Due to the impact of this parameter in the formulation, it is also shown in several papers that λ can also greatly affect the computational costs of branch-and-bound techniques when solving the optimization problem [6], [12].

The selection and evaluation of candidates is performed by the optimization algorithm, which seeks the minimization of the cost function. To solve this problem more efficiently, it is convenient to reformulate it as an equivalent BILS-problem. The necessary operations for this particular case are described in [6]. The final result is summarized in the expression:

$$\boldsymbol{U}_k^{\text{opt}} = \arg\min_{U_k} J_k = \arg\min_{U_k} \|\boldsymbol{H}\boldsymbol{U}_k - \bar{\boldsymbol{U}}_k^{\text{unc}}\|_2^2 \tag{2a}$$

$$\text{s. t. } \boldsymbol{U}_k \in \{0,1\}^{3N_p}, \tag{2b}$$

where (2b) are constraints that restrict the input sequences to members of the FCS extended for N_p. Also, $\boldsymbol{U}_k^{\text{unc}}$ is the unconstrained solution to the optimization problem. This can be computed every time step as $\boldsymbol{U}_k^{\text{unc}} = \boldsymbol{W}^{-1}\boldsymbol{F}_k$. Computation of matrices $\boldsymbol{W}$, $\boldsymbol{H}$ and vector $\boldsymbol{F}_k$ is described in [8]. Matrix $\boldsymbol{H}$ is a lattice generator which forms a $3N_p$-dimensional discrete space wherein the solution lies. The optimization problem is equivalent to finding the optimal $\boldsymbol{U}_k$ within the lattice with the shortest Euclidean distance J_k to $\bar{\boldsymbol{U}}_k^{\text{unc}} = \boldsymbol{H}\boldsymbol{U}_k^{\text{unc}}$ in the transformed space.

III. Sphere Decoding Algorithm

The conventional SDA for LPH-FCS-MPC applications is a branch-and-bound algorithm that can solve the BILS-problem. Its fundamental principle lies on the successive computation of candidate hyperspheres centered at the unconstrained solution $\bar{\boldsymbol{U}}_k^{\text{unc}}$. Each of these hyperspheres contains one candidate sequence from the FCS. It is guaranteed that candidates outside this hypersphere can be disregarded and eliminated from the search process. The SDA seeks to minimize the squared radius of the incumbent hypersphere until the smallest hypersphere is found, which corresponds to the optimal sequence $\boldsymbol{U}_k^{\text{opt}}$.

The evaluation of each hypersphere's squared radius can be successively performed following a sequential process in which each step considers a new component $\boldsymbol{U}_k(i)$. This process is usually illustrated with a tree diagram with $3N_p$ levels as shown in Fig. 2. Each level presents a predefined number of nodes that append a new individual switching position to the parent branch in component i. Thus, each node is a partial candidate $\boldsymbol{U}_k(i:3N_p-1) = \begin{bmatrix}\boldsymbol{U}_k(i) & \boldsymbol{U}_k(i+1) & \dots & \boldsymbol{U}_k(3N_p-1)\end{bmatrix}^T$. The conventional SDA follows a depth-first search strategy where vertical progression to an inferior layer $i-1$ is prioritized. However, when exploring a branch, if the partial cost exceeds that of the incumbent candidate, it is guaranteed that the subsequent nodes growing from this branch will not yield an optimal solution. Thus, these children nodes are not explored. This is done by performing a sidetracking movement, for which, a different switching position $\boldsymbol{U}_k(i)$ is assessed for the current tree layer i. If all the individual switching positions are explored from a parent node and none of them yield a better candidate than the incumbent, the branch can be pruned, effectively deleting all the subsequent children nodes from the search space. This is done by performing a backtracking movement, where the next layer corresponds to the parent node $(i+1)$. The imposed search strategy does not allow the algorithm to return to the pruned branches, remaining outside of the algorithm's scope.

In each node, a partial cost is computed and accumulated to the branch cost. The $3N_p+1$ dimensional array, $\boldsymbol{d}$ is used to calculate, store and retrieve these costs following the equation:

$$\boldsymbol{d}(i) = \|\boldsymbol{H}(i,:)\boldsymbol{U}_k - \bar{\boldsymbol{U}}_k^{\text{unc}}(i)\|_2^2 + \boldsymbol{d}(i+1), \tag{3}$$

where $\boldsymbol{H}(i,:)$ denotes row i of matrix $\boldsymbol{H}$ and $\boldsymbol{d}(3N_p) = 0$.

The SDA provides a certificate of optimality with important computational costs reductions in comparison with exhaustive enumeration. This feature has allowed practical application of several LPH-FCS-MPC works [5]–[7], [12]–[14]. Nonetheless, there is still an important drawback to the SDA method which is its inherently high computational variability. For this reason, it is necessary to introduce computational upper bounds that limit the amount of explored nodes so that a solution can be available within the allowed time, albeit suboptimal [28]. This implies that the SDA search stage may end prematurely. Thus, losing the certificate of optimality and introducing a certain degree of suboptimality that can noticeably degrade performance of LPH-FCS-MPC. This problem has been extensively studied and analyzed in the literature, leading to the proposal of several design

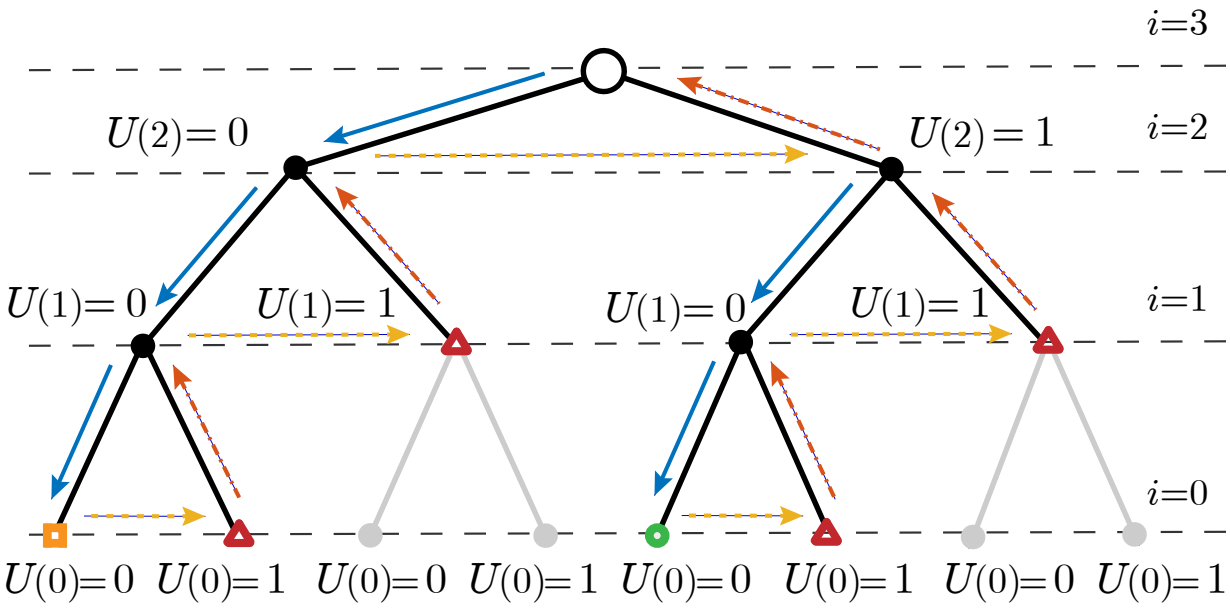


Fig. 2. Three-layer SDA search tree ($N_p = 1$). The search pattern is illustrated by arrows. Blue color indicates branches further explored to a deeper level ($i-1$). Yellow dashed arrows represent a horizontal step ($\boldsymbol{U}_k(i)+1$). Red dash-dotted arrows imply backtracking ($i+1$). Red triangle nodes indicate branch pruning. A yellow square indicates a suboptimal solution that was selected as incumbent. The green circle is the optimal solution.

and implementation techniques that address this computational concerns to achieve long N_p [6], [7], [12]–[14]. However, these proposals present a considerable drawback due to the extensive usage of FPGA resources and the development of hardware designs that are complex in terms of design and verification effort. For this reason, it is proposed to design an ANN that will learn the ideal SDA behavior without computational constraints. The ANN must be able to map the correct solution to each optimization problem in a more computationally efficient manner than the SDA.

IV. Artificial Neural Network for SDA Imitation

ANNs are mathematical models inspired by the human brain. Each neuron in the network provides a non-linear function, called activation function, of the sum of its weighted biased inputs [29]. Neurons are usually organized in layers, so the same inputs simultaneously feed all the neurons in the layer. In a deep ANN, layers are stacked. Thus, the outputs of each layer are connected to the next layer inputs. The supervised learning of ANN aims to establish the input-output mapping and functional relationships of complex problems such as the BILS-problem previously formulated [30]. Thus, the optimal sequence provided by the SDA algorithm is equivalent to class in a classification problem.

The ANN for MIMO detectors presented in [23] cannot be directly used in LPH-FCS-MPC due to the inherent differences of both applications. In the case of MIMO detectors, the terms in equations (2a) and (2b) are complex and different sets of constellations are usually available for transmitting the information (the equivalent of the FCS $\mathbb{V}$ in FCS-MPC). The main difference is related to the $\boldsymbol{H}$ matrix, which represents the MIMO channel: it is unknown by the receiver and it is usually time varying in mobile communications. For these reasons, a new ANN specifically designed for the SDA used in LPH-FCS-MPC is proposed.

A. Proposed ANN

The proposed ANN is a 3-layer neural network with 2 hidden layers, as shown in Fig.3, where l denotes the layer number. Thus, $l = \{1, 2\}$ are the hidden layers, and $l = 3$ is the output layer. A previous pre-processing block is added to both remove the mean value and to scale the ANN input to unit variance.

The inputs of the ANN (commonly called features) are the $M = 3N_p$ components of the unconstrained solution $\boldsymbol{U}^{\text{unc}}$, without transforming to the space generated by $\boldsymbol{H}$. This reduces the number of online calculations. The number of neurons in the hidden layers is also M, whereas the number of neurons in the output layer was chosen based on the number of switching states of the inverter (8). Some previous simulations showed that neither more neurons per layer nor more hidden layers improve the accuracy results.

Each inverter switching state is 8-bit one-hot coded, i.e, each state is represented by an 8-bit number where only one bit is set to “1” and the remaining bits are “0”. Regarding the ANN outputs, the neurons of the output layer encode the switching

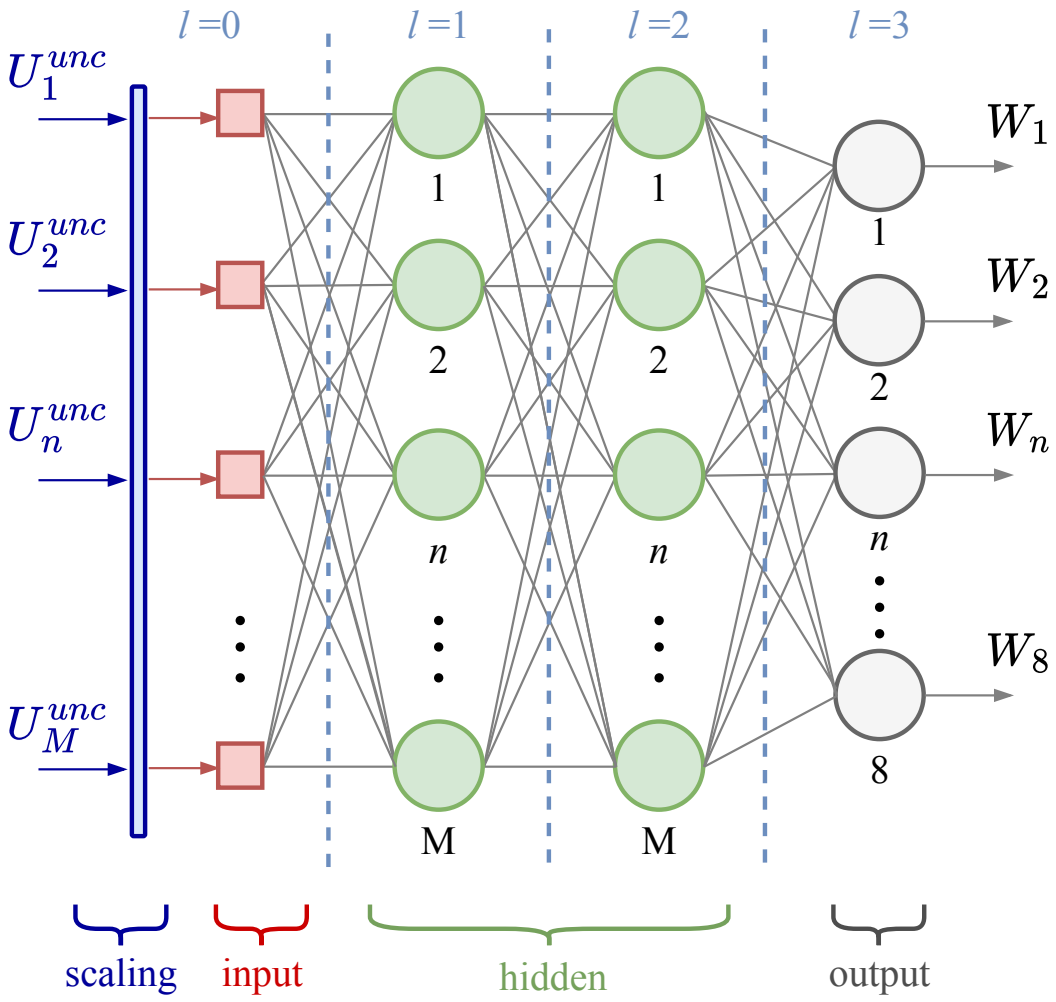


Fig. 3. Proposed ANN architecture. The inputs are normalized before feeding the ANN, where M neurons compose the hidden layers ($l = \{1, 2\}$), the same number as the inputs, and 8 neurons compose the output layer ($l = 3$).

state of the inverter by means of a softmax function, which is commonly used in other classification applications [31]. That is, each of the eight outputs neurons W_n provides a value between 0 and 1, with a total sum equal to 1. The higher W_n value is the one-hot coded version of the inverter switching state. The equivalent 3-bit coded version of this state is mapped to the three first components of the optimal sequence $\boldsymbol{U}_k^{\text{opt}}$ during the training process. The softmax function is only implemented during the training process because it is required by the categorical crossentropy function used as loss function [31]. In the hardware implementation, the softmax function is not performed, only its argument is calculated in each output neuron of the ANN. The resulting switching state is calculated by detecting the maximum of these arguments, since this maximum yields the same result and avoids calculating the softmax function. Regarding the activation function, a reduced complexity approximation of the hyperbolic tangent function shown in (4) is selected for all the neurons of the hidden layers.

$$f_a(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } x \in [-1, 1] \\ -1 & \text{if } x < -1 \end{cases} \tag{4}$$

ANN with ideal hyperbolic tangent was also tested and implemented, showing that the approximation has a negligible impact on the performance of the ANN, while reducing computational costs in comparison to the ideal.

B. Datasets and ANN training

Datasets were generated by simulation of the UPS model whose parameters are shown in Table III, which includes an ideal implementation of the SDA algorithm, i.e., a certificate of optimality is ensured. In order to cover several operation conditions and to improve the generalization of the proposed ANN, the models are set with the parameters shown in Table IV. That is, each

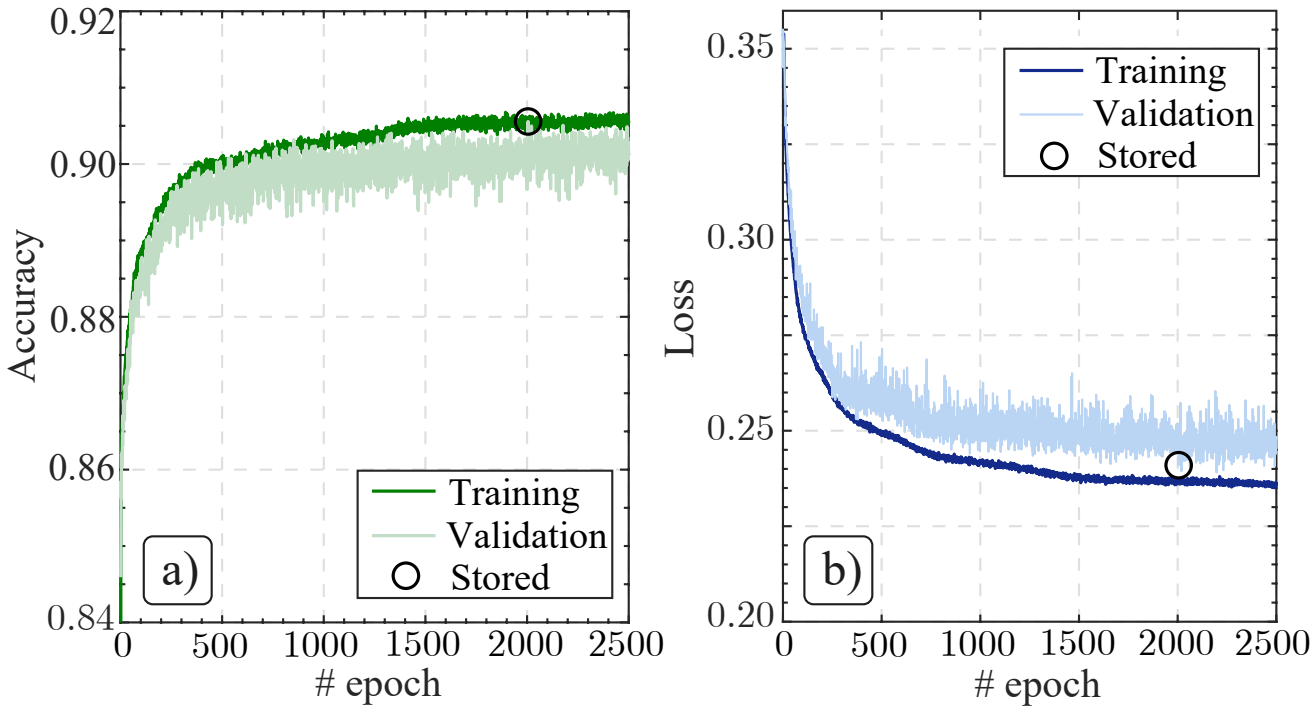


Fig. 4. Example of training process ($f_s = 25\,\text{kHz}$ and $N_p = 7$). a) accuracy, and b) loss function value on the validation dataset versus number of epoch.

TABLE III
UPS PARAMETERS

Parameter	Value
Output filter inductor, L_f	2 mH
Output filter capacitance, C_f	50 μF
DC-link voltage, v_{dc}	700 V
Voltage reference (phase to ground)	230 V
Fundamental frequency	50 Hz

TABLE IV
PARAMETERS VARIATIONS FOR THE TRAINING OF THE ANN

Parameter	Values
Load resistor [Ω]	$\{15, 30, 60\}$
Load inductor [mH]	$\{10, 20\}$
Target switching frequency, F^*_{sw} [kHz]	$\{1.5, 2, 2.5\}$

dataset includes simulations using three load resistors, two load inductors, and three target switching frequency values. One dataset was generated per each sampling frequency $f_s = \{20, 25, 33, 40\}$ kHz and per prediction horizon $N_p = \{1, 2, \ldots, 7\}$.

The equivalent time duration of the datasets were 20 periods of the fundamental frequency. By default, these datasets are unbalanced as the switching states are not uniformly distributed. Particularly, zero vectors are more probable than active vectors. To avoid the model to be biased to any switching state, the datasets were randomly down-sampled to the minority one. The resulting balanced dataset was splitted for training (80 %) and testing (20 %). In addition, the 20 % of the samples of each training dataset was reserved for model validation during the training process. The exact number of samples used for training depends on the trained case. For example, 88909 samples were used to train the model for configuration $f_s = 25\,\text{kHz}$ and $N_p = 7$.

The ANN model is implemented, trained, and evaluated using Keras, the deep learning API written in Python running on top of the machine learning platform TensorFlow. The interested reader is referred to [32] for a detailed description of the followed steps. For training, a variation of the stochastic gradient descent method called Adam optimizer has been selected [31]. A learning rate of 10^{-3}, batch size of 64 samples and 5000 epochs were selected to carry out the training. In some

TABLE V
LOSS FUNCTION / ACCURACY VALUES ON THE TEST DATA FOR DIFFERENT SAMPLING FREQUENCY f_s AND HORIZON (N_p).

N_p	$f_s = 20\,\mathrm{kHz}$	$f_s = 25\,\mathrm{kHz}$	$f_s = 33\,\mathrm{kHz}$	$f_s = 40\,\mathrm{kHz}$
1	0.051 / 0.98	0.036 / 0.99	0.031 / 0.99	0.033 / 0.99
2	0.077 / 0.97	0.075 / 0.97	0.048 / 0.98	0.041 / 0.99
3	0.148 / 0.94	0.119 / 0.95	0.073 / 0.97	0.071 / 0.97
4	0.225 / 0.91	0.200 / 0.92	0.094 / 0.96	0.079 / 0.97
5	0.217 / 0.91	0.187 / 0.93	0.140 / 0.94	0.114 / 0.95
6	0.283 / 0.88	0.216 / 0.91	0.154 / 0.94	0.119 / 0.95
7	0.261 / 0.90	0.242 / 0.90	0.177 / 0.93	0.137 / 0.95

TABLE VI
COMPUTATIONAL COMPLEXITY PER ANN LAYER: '+' AND '*' ARE ADDITIONS AND PRODUCTS, RESPECTIVELY.

Layer	$*$	$+$	Layer	$*$	$+$
0	$3N_p$	$3N_p$	2	$9N_p^2$	$3N_p$
1	$9N_p^2$	$3N_p$	3	$24N_p$	8

TABLE VII
MEASURED $\bar{F}_{sw}$ FOR $N_p = 7$. $\bar{F}_{sw}^*$ IS THE TARGET $\bar{F}_{sw}$ IN kHz. EACH PAIR REPRESENTS $\bar{F}_{sw}$ ANN / $\bar{F}_{sw}$ IDEAL SDA

$\bar{F}_{sw}^*$	$f_s = 20\,\mathrm{kHz}$	$f_s = 25\,\mathrm{kHz}$	$f_s = 33\,\mathrm{kHz}$	$f_s = 40\,\mathrm{kHz}$
1.5	1.57 / 1.55	1.55 / 1.54	1.47 / 1.49	1.65 / 1.68
2.0	2.13 / 2.06	1.99 / 2.01	1.94 / 2.00	1.97 / 1.97
2.5	2.50 / 2.49	2.55 / 2.50	2.40 / 2.52	2.47 / 2.53

cases the batch size is increased up to 128 to avoid convergence problems. As other classification problems, the categorical cross-entropy function is used as loss function [31].

To avoid over-fitting, the following measures have been implemented: (a) the hidden layers include L1 and L2 regularization, and (b) validation-based early-stopping. L1-L2 regularization includes a term in the cost function that penalizes the network weights [31], whereas early-stop uses the accuracy on the validation set to premature stop the training processes. If the validation accuracy does not improve in the patience time of 500 epochs, the training process is stopped, and the weights are restored from the best epoch in that set [31].

To illustrate the training process, the accuracy and loss function histories for case $N_p = 7$ and $f_s = 25\,\mathrm{kHz}$ are shown in Fig. 4 (a) and (b), respectively. The monotonous evolution of both, the results obtained with training and validation datasets, with no sign of divergence, shows that the over-fitting has been avoided. The accuracy shown in Fig. 4(a) exhibits that the training loop is stopped at epoch 2500, after 500 epochs with no improvement.

C. Trained models results

Table V shows the accuracy and the value of the loss function. Lower horizons and higher sampling frequency exhibit better performances for both, loss function and accuracy.

The confusion matrix of case $N_p = 7, f_s = 25\,\mathrm{kHz}$ is shown in Fig. 5. Here, each element represents the instances of known versus predicted switching states, both coded in decimal. The bottom row and the column on the right summarize the information of columns and rows, respectively. The lower right corner shows the total accuracy of the model. Note that, switching states 0 and 7 (the same switching state) exhibit lower accuracy. In addition, the predicted switching states are uniformly distributed (see the right column of Fig. 5), indicating that the ANN does not alter the distribution of the switching states provided by the SDA (see bottom row of Fig. 5).

Regarding computing complexity, Table VI shows the number of arithmetic operations that the proposed ANN requires for a given horizon N_p.

V. PERFORMANCE EVALUATION

A. Simulation Results

The first analyses use the same data sets and simulation model employed in Subsection IV-B for the training of the ANN. Unless otherwise specified, all the results presented in this subsection have been obtained simulating 20 fundamental cycles and averaging the results for all the combinations of parameters shown in Table IV.

The simulation analysis focuses on the optimization process performance and how this translates to the control outcome. For this, two metrics will be the focus of the study: optimality $Op(\%)$ and total harmonic distortion THD$(\%)$. Optimality refers to the optimization phase success rate. An ideal SDA is capable of unequivocally finding the optimal, i.e. provides an

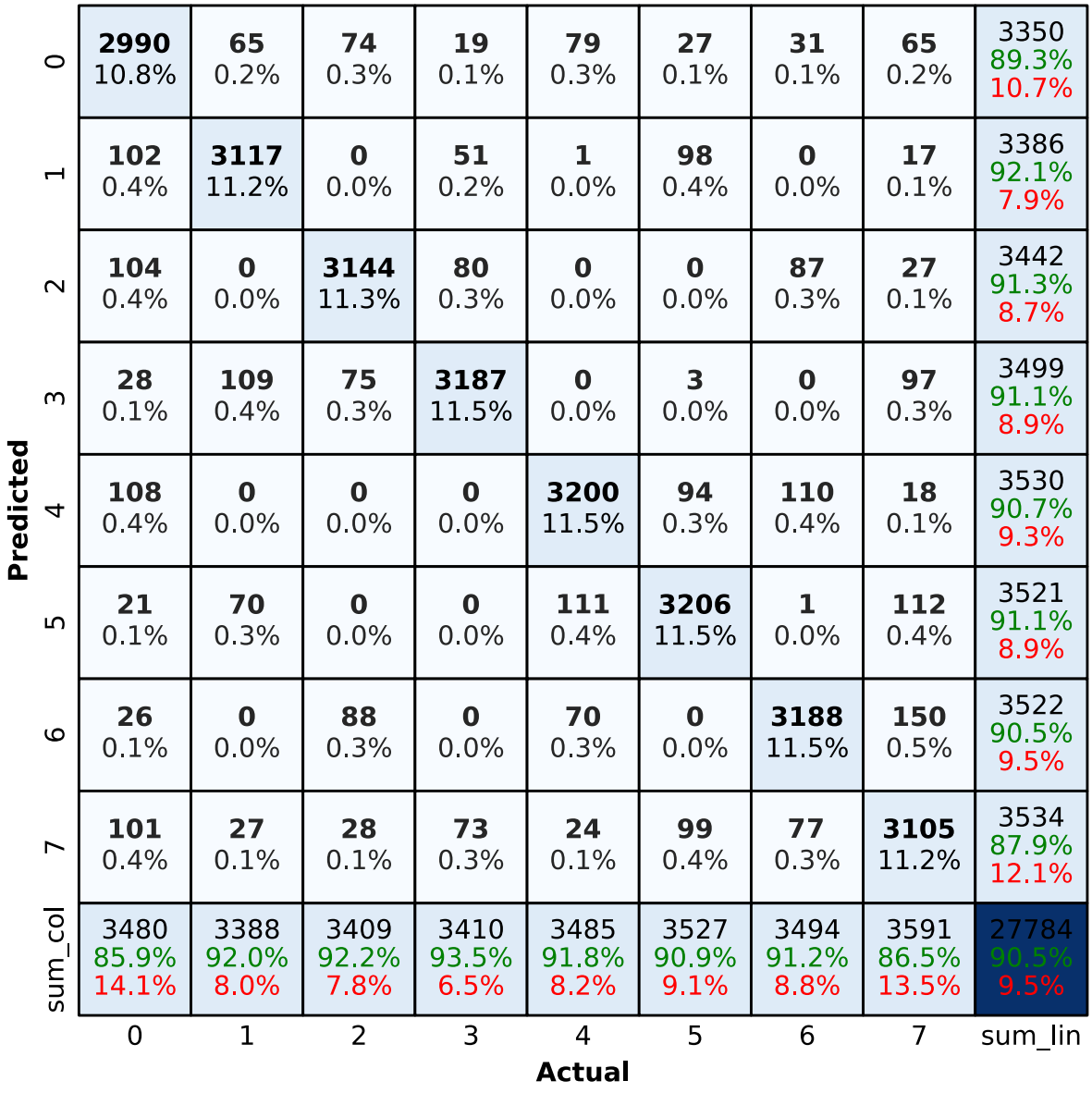


Fig. 5. Confusion matrix for case: $f_s = 25\,\mathrm{kHz}$ and $N_p = 7$. It is evaluated only on the dataset test portion.

optimality of 100 %. However, an ANN or a constrained SDA algorithm may select suboptimal solutions in some occasions. In simulation, it is possible to compute a percentage of successfully found optimal solutions in the total amount of optimization problems computed in the considered time frame. As a byproduct of this suboptimality, a certain control outcome degradation is to be expected. To address control outcome in the analyzed UPS application, the THD of the output voltage is computed and represented.

Fig. 6a shows the optimality of the proposed ANN architecture for different N_p and f_s values. As stated in Subsection IV-B, each point of the graphs corresponds to a particular ANN implementation trained for these particular pair of values. From this figure it can be seen that for the depicted range of N_p and f_s the optimality is always greater than 85%. As seen in Subsection IV-C, lower prediction horizon lengths and higher sampling frequencies exhibit better optimality. Also, suboptimal solutions selected by the ANN-SDA have a low impact in terms of THD degradation. Fig. 6b also shows the evolution of the THD versus N_p, comparing the ideal SDA algorithm (with no limit in the number of explored nodes) and the proposed ANN architecture for different values of f_s. From this figure, it can be seen that the THD obtained with the ANN is very similar to the one with the ideal SDA. The greater difference occurs at $f_s = 25$ kHz and $N_p = 6$ where it is lower than 0.2%.

Regarding the switching frequency, Table VII shows the measured switching frequency versus the sampling frequency, comparing the proposed ANN architecture and the ideal SDA, for different target switching frequencies. From this table, it can be seen that the ANN can imitate the outcome of the ideal SDA while yielding similar switching effort.

Additionally, the generalization ability of the designed ANN was evaluated, i.e., its ability to adapt properly to new and

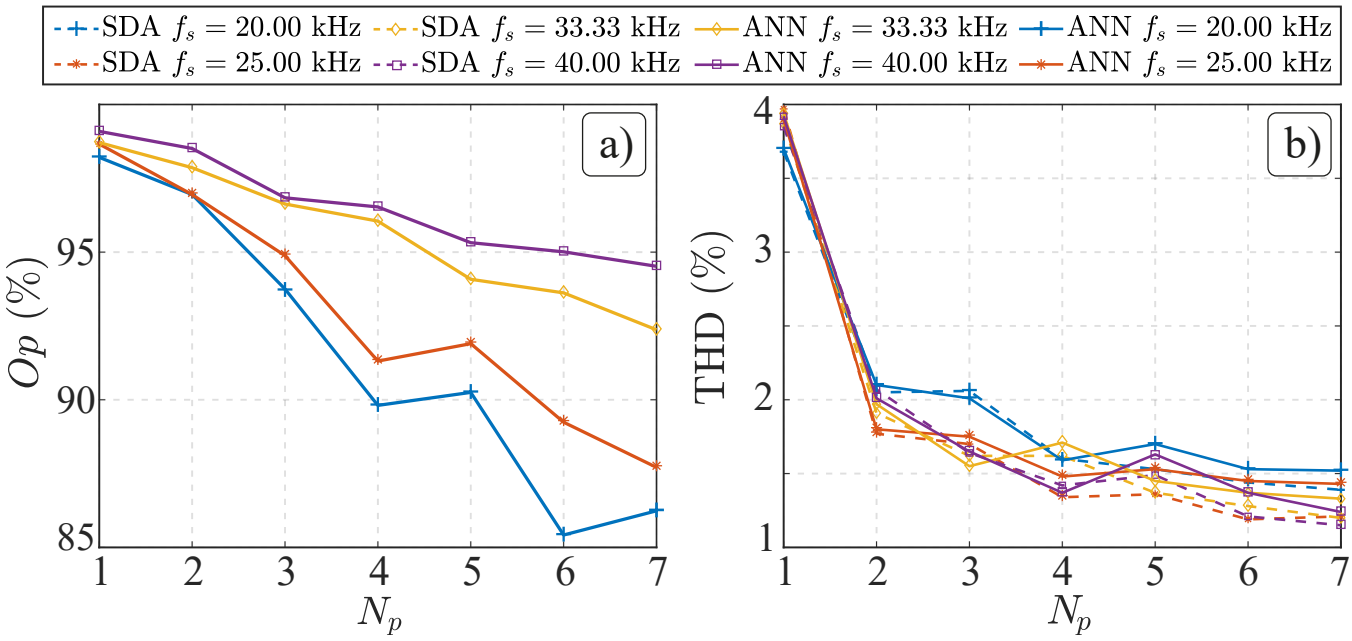


Fig. 6. a) Optimality versus N_p. b) THD versus N_p for different f_s values. The THD obtained with the ideal SDA is also shown for comparison.

TABLE VIII
EXECUTION TIMES FOR ANN-BASED LPH-FCS-MPC ALGORITHM.

Ex. Time	N_p =3	N_p =4	N_p =5	N_p =6	N_p =7
T_{adc}			7.53 μs		
T_{pre}	3.11 μs	4.30 μs	5.85 μs	7.56 μs	10.09 μs
T_{ANN}	3.76 μs	5.42 μs	7.33 μs	9.75 μs	12.21 μs

previously unseen data. Simulations with loads different than the ones used in the training process were tested, more specifically, load resistors of $\{10, 20, 40, 50, 70\}\Omega$, and load inductors of $\{5, 15, 25\}$mH. The obtained results, not presented here for the sake of brevity, are almost the same than the one presented in Fig. 6. For instance, for a target switching frequency of 2 kHz, the absolute value of the differences between the THD obtained with the new loads and the one shown in Fig. 6 are always lower than 0.08%. These results prove that the ANN is able to generalize well for new load values.

B. Experimental Results

Experimental tests are performed in an UPS laboratory prototype that consists of a 20-kVA 2L-VSI connected to an LC filter. The VSI is composed by three SKM100GB12T4 commercial IGBT modules from Semikron operated using the Semikron 32 PRO R gate driver. Parameters are the same as in the simulation study unless otherwise specified. The utilized load is a 30 Ω resistor with a 20 mH inductor. The digital control is implemented in a low-cost PYNQ-Z1 development board from Digilent Inc. The board is based on a Xilinx Zynq-7000 FPSoC. Execution times for the entire algorithm are depicted in Table VIII in terms of ADC conversion time, preliminary calculations and ANN computation.

Four optimization techniques are assessed in the comparisons. The first method is the proposed ANN-SDA design executed in one ARM core. The second method is the conventional constrained SDA, as proposed in [8], [14] and executed in the same ARM core. For further validation, the FPGA-based, parallel SDA proposed in [12] and K-best SDA proposed in [6] are also included in the comparative results.

For these tests, the parallel SDA is configured with 8 parallel blocks and K-best SDA with $K_b = 8$. Harmonic distortion results for different parameters are depicted in Fig. 7. As can be seen, the proposed ANN-SDA design provides a vastly superior performance to the conventional SDA. In particular, the conventional SDA implemented in microprocessor yields a very degraded performance for $N_p > 4$ due to the required computational upper bounds. For this application, it is found that the maximum number of nodes that can be explored halves by each incremental unit of N_p. For instance, this value is 60 nodes for N_p=5 and 30 nodes for N_p = 6, which is insufficient for proper solution of the optimization problem. In contrast, the ANN-SDA achieves further improvements as N_p is increased. Also, performance is generally better than other advanced techniques that require more complex and resource demanding implementations, as they similarly suffer from computational restrictions. This fact is demonstrated in the experiment shown in Fig. 8. As can be seen, when selecting a prediction horizon of

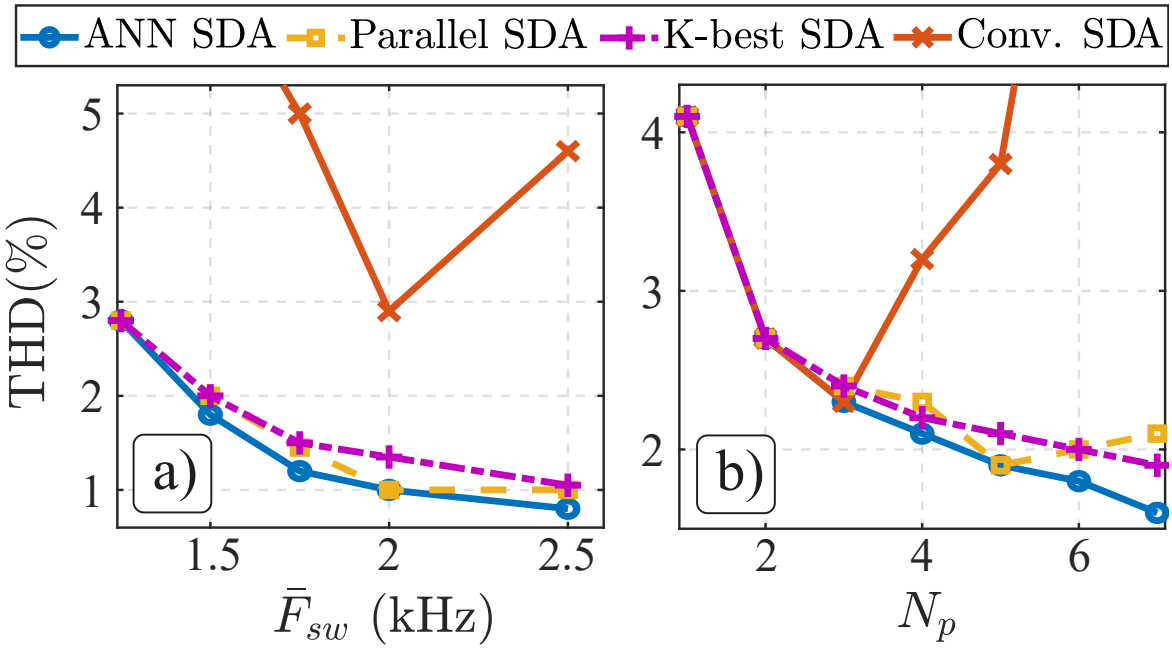


Fig. 7. Steady-state performance as a measure of the output voltage THD. Sampling frequency is f_s = 25 kHz. a) Switching frequency sweep for N_p = 6. b) Prediction horizon sweep for $\bar{F}_{sw}$ = 1.5 kHz.

$N_p = 7$ and $\bar{F}_{sw}$ = 2 kHz, the optimization problem is too complex for parallel SDA and K-best SDA methods. However, the ANN-SDA has been able to properly learn the optimization problem from the ideal SDA and provide more accurate solutions.

Another important topic regarding the ANN-SDA method is adaptability to operation points outside the training set. For instance, in the proposed design, training is performed for a set of three different switching frequency values $\bar{F}_{sw} = \{1.5, 2, 2.5\}$ kHz, which correspond to specific parameter λ values depending on parameters such as N_p or f_s among others. Even if training is performed for these specific λ values, the proposed method decouples the optimization phase from the rest of the control algorithm. Thus, in contrast to other ANN-based methods fin the literature such as [20], [21], it is possible to tune different λ parameters that affect the calculation of the unconstrained solution. Interestingly, experimental results highlight that the proposed ANN-SDA provides great flexibility to parameters outside the training set. This fact is depicted in Fig. 9.

VI. Conclusion

The main contribution of this paper is the design and development of an ANN imitator for the conventional SDA to solve the underlying optimization problem in LPH-FCS-MPC strategies. Solving the optimization stage with an ANN provides a method with fixed computational costs where a great part of the problem original computational burden is shifted to the offline training. The results highlight the validity and the benefits of this method as the achieved performance is vastly superior to the conventional SDA executed in the same microprocessor with computational upper bounds. The proposed method is also compared against more advanced optimization techniques in the literature that require costly hardware resources. The proposed ANN-SDA can outperform these techniques while maintaining a strong robustness for parameters outside the training data. For this reason, ANN-based SDA has been revealed as a very promising tool to help the growth of LPH-FCS-MPC strategies and enable wider application in the industry.

References

[1] P. Karamanakos and T. Geyer, "Guidelines for the design of finite control set model predictive controllers," *IEEE Transactions on Power Electronics*, vol. 35, DOI 10.1109/TPEL.2019.2954357, no. 7, pp. 7434–7450, 2020.

[2] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open Journal of Industry Applications*, vol. 1, DOI 10.1109/OJIA.2020.3020184, pp. 95–114, 2020.

[3] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, DOI 10.1109/TIE.2016.2625238, no. 2, pp. 935–947, Feb. 2017.

[4] C. Bordons and C. Montero, "Basic principles of mpc for power converters: Bridging the gap between theory and practice," *IEEE Industrial Electronics Magazine*, vol. 9, DOI 10.1109/MIE.2014.2356600, no. 3, pp. 31–43, Sep. 2015.

[5] P. Acuna, C. A. Rojas, R. Baidya, R. P. Aguilera, and J. E. Fletcher, "On the impact of transients on multistep model predictive control for medium-voltage drives," *IEEE Transactions on Power Electronics*, vol. 34, no. 9, pp. 8342–8355, 2019.

[6] E. Zafra, S. Vazquez, A. M. Alcaide, L. G. Franquelo, J. I. Leon, and E. P. Martin, "K-best sphere decoding algorithm for long prediction horizon fcs-mpc," *IEEE Transactions on Industrial Electronics*, vol. 69, DOI 10.1109/TIE.2021.3104600, no. 8, pp. 7571–7581, 2022.

[7] E. Liegmann, P. Karamanakos, and R. Kennel, "Real-time implementation of long-horizon direct model predictive control on an embedded system," *IEEE Open Journal of Industry Applications*, vol. 3, DOI 10.1109/OJIA.2021.3133477, pp. 1–12, 2022.

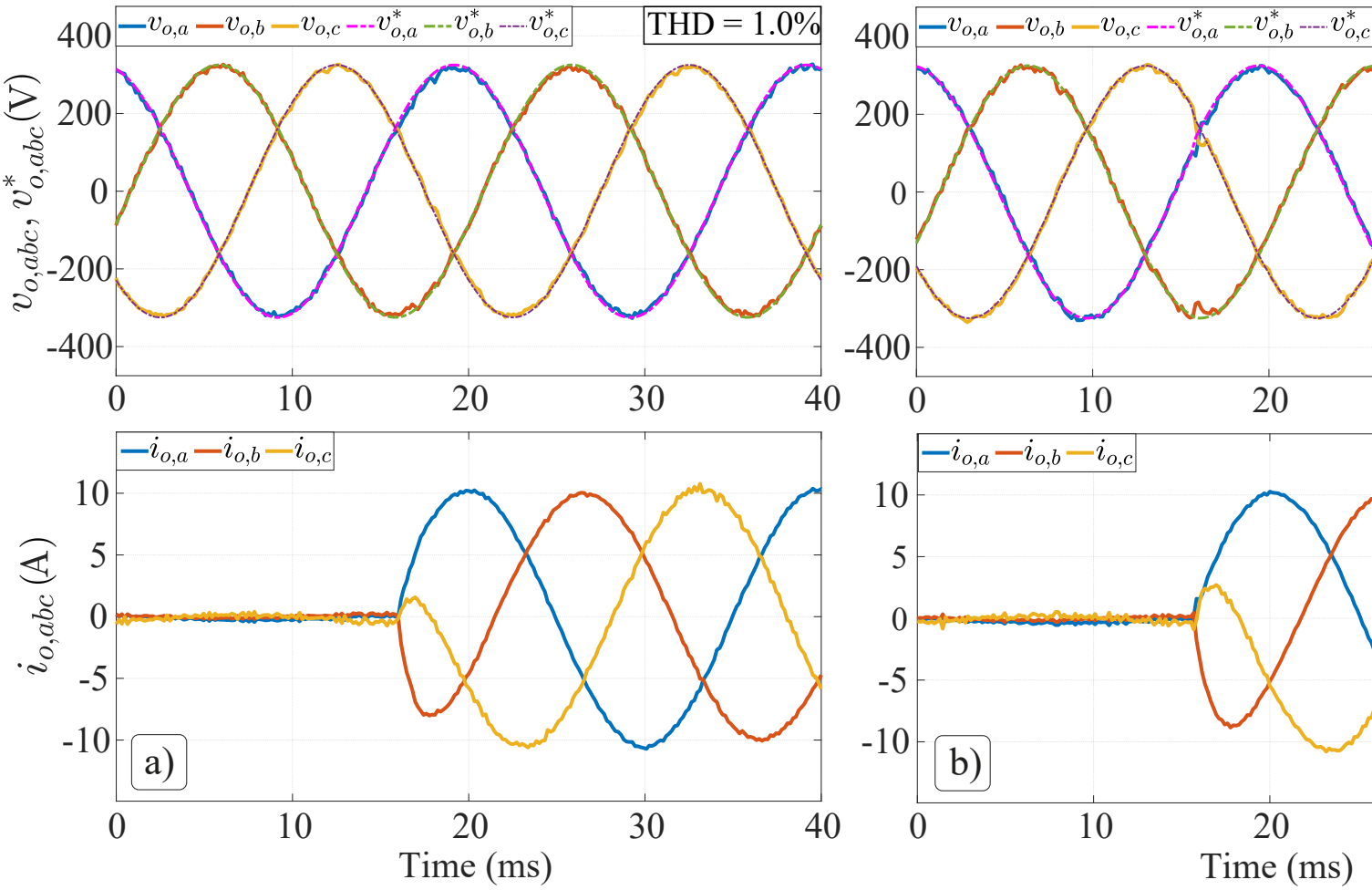


Fig. 8. Experimental results for N_p = 7, f_s = 25 kHz and $\bar{F}_{sw}$ = 2 kHz. Three phase magnitudes are represented. Top figures show the output voltage and reference. Bottom figures show the output current. a) ANN-SDA. b) Parallel SDA with 8 parallel blocks. c) K-best SDA with K_b=8.

[8] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6836–6846, 2014.

[9] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1633–1644, 2015.

[10] R. Baidya, R. P. Aguilera, P. Acuna, S. Vazquez, and H. d. T. Mouton, "Multistep model predictive control for cascaded h-bridge inverters: Formulation and analysis," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 876–886, 2018.

[11] R. Baidya, R. P. Aguilera, P. Karamanakos, P. Acuna, C. Rojas, T. Geyer, and D. D. . Lu, "Dealing with suboptimality in multistep model predictive control for transient operations," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 3780–3785, 2019.

[12] E. Zafra, S. Vazquez, C. Regalo, V. B. Lecuyer, A. M. Alcaide, J. I. Leon, and L. G. Franquelo, "Parallel sphere decoding algorithm for long-prediction-horizon fcs-mpc," *IEEE Transactions on Power Electronics*, vol. 37, DOI 10.1109/TPEL.2022.3150433, no. 7, pp. 7896–7906, 2022.

[13] E. Zafra, S. Vazquez, A. M. Alcaide, E. P. Martin, L. G. Franquelo, and J. I. Leon, "Hybrid sphere decoder for long prediction horizon fcs-mpc," *IEEE Transactions on Industrial Electronics*, vol. 70, DOI 10.1109/TIE.2022.3194587, no. 6, pp. 5484–5492, 2023.

[14] T. Dorfling, H. du Toit Mouton, T. Geyer, and P. Karamanakos, "Long-horizon finite-control-set model predictive control with nonrecursive sphere decoding on an fpga," *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7520–7531, 2020 .

[15] S. A. Bin Khalid, E. Liegmann, P. Karamanakos, and R. Kennel, "High-level synthesis of a long horizon model predictive control algorithm for an fpga," in *PCIM Europe digital days 2020*, pp. 1–8 , 2020.

[16] T. Dragičević and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Transactions on Industrial Electronics*, vol. 66, DOI 10.1109/TIE.2018.2875660, no. 11, pp. 8870–8880, 2019.

[17] M. Novak, H. Xie, T. Dragičević, F. Wang, J. Rodriguez, and F. Blaabjerg, "Optimal cost function parameter design in predictive torque control (ptc) using artificial neural networks (ann)," *IEEE Transactions on Industrial Electronics*, vol. 68, DOI 10.1109/TIE.2020.3009607, no. 8, pp. 7309–7319, 2021.

[18] S. Vazquez, D. Marino, E. Zafra, M. D. V. Peña, J. J. Rodriguez-Andina, L. G. Franquelo, and M. Manic, "An artificial intelligence approach for real-time tuning of weighting factors in fcs-mpc for power converters," *IEEE Transactions on Industrial Electronics*, vol. 69, DOI 10.1109/TIE.2021.3127046, no. 12, pp. 11 987–11 998, 2022.

[19] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragičević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output lc filter," *IEEE Access*, vol. 7, DOI 10.1109/ACCESS.2019.2938220, pp. 124 737–124 749, 2019.

[20] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Long-horizon direct model predictive control based on neural networks for electrical drives," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, DOI 10.1109/IECON43393.2020.9254388, pp. 3057–3064, 2020.

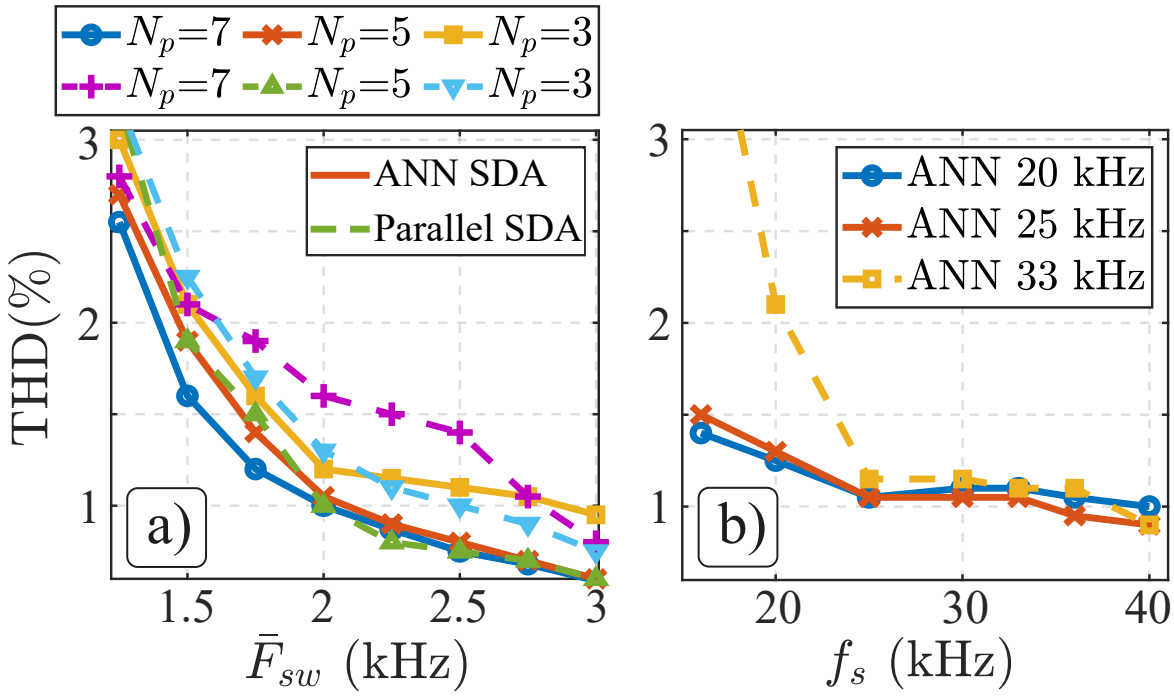


Fig. 9. ANN-SDA performance for different parameters. a) THD for different N_p and $\bar{F}_{sw}$. b) THD for ANN-SDA at $N_p = 5$ working at sampling frequencies different than the training value (ANN x kHz).

[21] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Transactions on Industrial Electronics*, vol. 68, DOI 10.1109/TIE.2020.2969116, no. 2, pp. 1717–1723, 2021.

[22] Proakis, *Digital Communications 5th Edition*. McGraw Hill, 2007.

[23] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, DOI 10.1109/TSP.2019.2899805, no. 10, pp. 2554–2564, 2019.

[24] N. T. Nguyen and K. Lee, "Deep learning-aided tabu search detection for large mimo systems," *IEEE Transactions on Wireless Communications*, vol. 19, DOI 10.1109/TWC.2020.2981919, no. 6, pp. 4262–4275, 2020.

[25] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive mimo," *IEEE Transactions on Wireless Communications*, vol. 19, DOI 10.1109/TWC.2020.2996144, no. 8, pp. 5635–5648, 2020.

[26] J. Sun, Y. Zhang, J. Xue, and Z. Xu, "Learning to search for mimo detection," *IEEE Transactions on Wireless Communications*, vol. 19, DOI 10.1109/TWC.2020.3012785, no. 11, pp. 7571–7584, 2020.

[27] S. Vazquez, E. Zafra, R. P. Aguilera, T. Geyer, J. I. Leon, and L. G. Franquelo, "Prediction model with harmonic load current components for fcs-mpc of an uninterruptible power supply," *IEEE Transactions on Power Electronics*, vol. 37, DOI 10.1109/TPEL.2021.3098948, no. 1, pp. 322–331, 2022.

[28] P. Karamanakos, T. Geyer, and R. Kennel, "Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 334–341, 2015.

[29] S. Haykin, *Neural Networks and Learning Machines*, ser. Neural networks and learning machines, no. v. 10. Prentice Hall, 2009. [Online]. Available: https://books.google.es/books?id=K7P36lKzI_QC

[30] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, vol. 36, DOI 10.1109/TPEL.2020.3024914, no. 4, pp. 4633–4658, 2021.

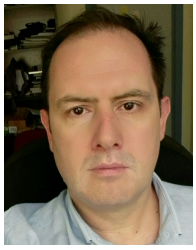
[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[32] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, 2019.

Eduardo Zafra (Student Member, IEEE) was born in Seville, Spain in 1994. He received the B.S. and M.S. degrees in industrial engineering from the University of Seville (US) in 2016 and 2019 respectively.

He is currently pursuing a PhD degree in automatics, electronics and telecommunications at University of Seville. His main research interests include control and modulation of power converters and drives, model predictive control and design for digital embedded systems.

Joaquin Granado was born in Cordoba (Spain) in 1973 and received his Master and Ph.D. degrees in Telecommunication Engineering in 1999 and 2005, both from the University of Seville. He is currently with the Electronic Engineering Department of this University. His research interests includes signal processing, modeling, measuring and instrumentation for a wide range of applications.



Vicente Baena Lecuyer was born in Athis-Mons, France. He received his Telecommunication Engineering and Ph.D. degrees from the University of Seville (Spain) in 1997 and 2001, respectively. Since 1997, he has been with the Department of Electronic Engineering, Escuela Superior de Ingenieros, University of Seville. His current research interest is mainly in digital design of communications systems.

Sergio Vazquez (Fellow, IEEE) was born in Seville, Spain, in 1974. He received the M.S. and PhD degrees in industrial engineering from the University of Seville (US) in 2006, and 2010, respectively.

Since 2002, he is with the Power Electronics Group working in R&D projects. He is an Associate Professor with the Department of Electronic Engineering, US. His research interests include power electronics systems, modeling, modulation, and control of power electronics converters applied to renewable energy technologies.

Dr. Vazquez was recipient as coauthor of the 2012 Best Paper Award of the IEEE Transactions on Industrial Electronics and 2015 and 2022 Best Paper Award of the IEEE Industrial Electronics Magazine. He is involved in the Energy Storage Technical Committee of the IEEE industrial electronics society and is currently serving as an Associate Editor of the IEEE Transactions on Industrial Electronics.

Abraham Marquez (Member, IEEE) was born in Huelva, Spain, in 1985. He received his B.S., M.S. and Ph.D degrees in telecommunications engineering from Universidad de Sevilla (US), Seville, Spain in 2014, 2016 and 2019, respectively. His main research interests are advanced modulation techniques, multilevel converters, modular converters, model-based predictive control of power converters and drives, renewable energy sources, thermal modeling of power converters and power devices lifetime extension. Dr. Marquez is coauthor of more than 40 journal papers, and participated in more than 10 R&D projects. Dr. Marquez was recipient as coauthor of the 2015 and 2021 Best Paper Award of the IEEE Industrial Electronics Magazine.

Jose I. Leon (Fellow, IEEE) was born in Cadiz, Spain. He received the B.S., M.S., and PhD degrees in telecommunications engineering from Universidad de Sevilla (US), Seville, Spain, in 1999, 2001, and 2006, respectively.

He is an Associate Professor with the Department of Electronic Engineering, US. Since 2019 he is also Chair Professor at the Department of Control Science and Engineering in Harbin Institute of Technology (China).

His research interests include modulation and control of power converters for high-power applications and renewable energy systems. Dr. Leon was a co-recipient of the 2008 Best Paper Award of IEEE Industrial Electronics Magazine, the 2012 Best Paper Award of the IEEE Transactions on Industrial Electronics, and the 2015 Best Paper Award of IEEE Industrial Electronics Magazine. He was the recipient of the 2014 IEEE J. David Irwin Industrial Electronics Society Early Career Award, the 2017 IEEE Bimal K. Bose Energy Systems Award and the 2017 Manuel Losada Villasante Award for excellence in research and innovation. In 2017 he was elevated to the IEEE fellow grade with the following citation ”for contributions to high-power electronic converters”.

Leopoldo G. Franquelo (Life Fellow, IEEE) was born in Malaga, Spain. Ph.D. in Industrial Engineering (1980) from the University of Seville, where he has been a professor in the Department of Electronic Engineering since 1986. His current research interests include modulation techniques for multilevel inverters and their application to renewable energy systems. He has participated in more than 100 R&D projects and has published more than 350 papers, 120 of them in high impact journals. Dr. Franquelo has been the IES vice president for conferences and president currently is a Distinguished Lecturer at the Industrial Electronics Society (IES) since 2006. Committed to the work in IEEE, he was Editor in Chief (2016-2019) of the IEEE Transactions on Industrial Electronics and is currently Editor in Chief of the IEEE Open Journal of the Industrial Electronics Society. In 2009 and 2013 he received the prestigious Andalusian Research Award and the FAMA Award in recognition of the excellence of his research career. In addition to other recognitions for his research work, he received the Eugene Mittelmann Outstanding Research Achievement Award and the Anthony J. Hornfeck Service Award from the IEEE-IES, in 2012 and 2015 respectively.