Electrical Power and Energy Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Pseudo-optimal five-level DCC modulation based on machine learning ☆

Pablo Montero-Robina [a,*], Francisco Gordillo [b], Fabio Gómez-Estern [c], Federico Cuesta [b]

[a] Autel Iberia, S.L., Barcelona, Spain
[b] Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain
[c] Escuela Técnica Superior de Ingeniería, Universidad Loyola Andalucía, Sevilla, Spain

ARTICLE INFO

Keywords:
Classification and regression trees
Diode-clamped converter
Mixed-integer linear optimization
Multilevel converter

ABSTRACT

This paper presents a method for the control design of five-level DCC converters based on mixed-integer optimization and machine learning. The resulting controller is computationally simple and can be easily implemented on low-resource control hardware using simple nested "if-else" statements. The optimization problem is recalled from previous work by modifying the cost function to further enhance the dynamic performance. Additionally, and in contrast to previous works, the online implementation accomplished in this paper allows the system to cover a wider range of operating points. For this, the optimization problem is solved offline for several operating conditions, and the results are gathered into a dataset to train classification and regression trees (CARTs), which are later used online. Due to the generalization capability of the CARTs, a more flexible and less resource-intensive implementation is achieved which is capable of operating at points outside the ones considered in the training dataset. The resulting control strategy is compared in simulation and experiments with several alternative approaches found in the literature. This approach can be extended to other power converter topologies, allowing the implementation of optimized modulations.

1. Introduction

Trends in energy consumption encourage researchers to look for feasible solutions that reduce the cost of grid infrastructure. High-power industrial applications have been the scope of this search, as they facilitate the delivery and consumption of energy from the high-voltage distribution grid in a straightforward way [1]. Therefore, the voltage limits of power electronics devices are a major concern in the field [2]. In this context, multilevel converter topologies emerged as a predominant solution [3]. The use of serialized switching devices enables them to support larger grid voltage by dividing it among the connected devices [4]. Furthermore, they can achieve more accurate modulation and reduction of current distortion, which facilitates the fulfillment of the grid codes EN50160. However, as the number of levels increases, so does the control complexity as there are more possible switching states. Besides, the additional capacitors have to be balanced, which involves additional control objectives. If this aspect is not taken into account, the system may not behave properly or even suffer irreversible damage.

There are a large number of multilevel converter topologies, and each has its advantages and drawbacks [5]. This paper focuses on the diode-clamped converter (DCC) [6]. The three-level neutral-point-clamped (NPC) converter belongs to this topology and is one of the most investigated, accepted, and used multilevel converter topologies in the industry. However, the five-level one still finds some reluctance. This is due to the increase in the number of switching vectors (125) and the control objectives that must be considered. Therefore, designing a proper modulation algorithm to command the switches is vital for the integration of this topology. The modulation must satisfactorily implement the desired output voltages at the same time that the capacitors must be evenly charged. This is not an easy task, and many different approaches are currently found in the literature as shown below.

A strategy based on space-vector modulation (SVM) that solves the modulation and balancing issues is the one presented in [7], which uses a combination of a small switching vector and two outer ones in such a way that balancing capabilities can be achieved using the redundant vectors of the small one. The work shown in [8] further takes this approach to produce a switching sequence that guarantees smooth transitions between switching periods. These approaches will be referred to as space-vector-based algorithms (SVBA) from now on for comparative purposes. In another line, [9] uses carrier-based PWM with zero-sequence injection in such a way that it takes into account the

☆ This work has been funded by the Spanish Ministry of Science, Innovation and Universities under grant PID2019-109071RB-I00/AEI/10.13039/501100011033 and by Junta de Andalucía-FEDER grant P20-01116.

* Corresponding author.
E-mail addresses: pmontero1@us.es (P. Montero-Robina), gordillo@us.es (F. Gordillo), fgestern@uloyola.es (F. Gómez-Estern), fcuesta@us.es (F. Cuesta).

https://doi.org/10.1016/j.ijepes.2023.109677
Received 30 October 2022; Received in revised form 12 August 2023; Accepted 20 November 2023

capacitor voltage balance for a five-level DCC. However, this approach only offers balancing capabilities for applications with low power factor values or low modulation indices, as operating with high modulation indexes does not guarantee capacitor voltage balancing in five-level DCC [8]. Alternatively, work [10] presents a modulation strategy that decouples the achievement of the output voltage from the capacitor voltage balancing control accomplishing both control objectives simultaneously. Thus, it is referred as integrated control modulation (ICM). For this, the modulation could use any level within a switching period. Furthermore, the original ICM (OICM) is modified (MICM) to improve performance. This strategy is based on the implementation of PWM for every level. Any of the aforementioned approaches present a solution for the modulation and voltage balancing of five-level DCC but they are not optimized in terms of number of commutations, that is, using the minimal amount of levels every switching period.

In the field of optimal approaches, one well-known option is the finite-control set model predictive control (FCS-MPC) [11]. For this, at every sampling interval, all possible switching states are evaluated with a cost function that covers, with different weights, the control objectives: current reference tracking, capacitor voltage balancing, and reduced number of commutations. Therefore, the optimum switching state is selected and implemented in the next sampling interval. Note that the performance of the system depends on how these objectives are weighted in the cost function. As a noticeable disadvantage, the FCS-MPC selects only one switching state in each sampling period, losing the advantage the PWM has of commuting inside the switching period. In [12], by defining a mixed-integer linear programming optimization problem, with control objectives as constraints and key variables to account for the optimal result, a cost function leads to the best solution in terms of duty ratios for every phase at every level. Therefore, in contrast to FCS-MPC, the PWM is still used, as the outputs of the optimization problem are the levels that will be used every switching period. Unfortunately, this optimization problem has a large computational burden and cannot be executed online. A solution was presented in [13], where the optimization problem was solved offline for an operating point under steady state conditions, and the optimization results were stored in a look-up table (LUT)—this approach will be referred to as LUT from now on. However, large variations in the modulation index or the power factor can change the conditions under which the LUT was obtained, and hence impede the fulfillment of the control objectives. Consequently, the LUT approach seems infeasible for systems that require a wider range of operating points. To overcome this and to avoid the need to solve the optimization problem at every sampling instant, we propose in this paper a machine learning-based approach, with improved optimization, that deals with this lack of generality and difficulty of online implementation in such a way that the resulting online algorithm takes the form of decision trees, which are simple and require very low computational load. In summary, an easy-to-implement algorithm is obtained that seeks optimality while still using PWM.

Machine learning is probably the fastest-spreading technical field today, boosted by the steady increase of processing power, the abundance of data thanks to mobile and IoT technologies, and the refinement of data science algorithms. In automatic control, neural networks have been the most common machine learning tool since the 1990s [14]. whether as a predictor, a tool for optimization, decision making, or determination of the control action. Machine learning algorithms have also been used for a long time in converter control applications [15,16]. In a recent paper [17], real-time implementation of DC/DC power converter control based on deep machine learning techniques is analyzed. See [18], and the references therein, for a detailed overview of artificial intelligence applications for power electronics, including expert systems, fuzzy logic, metaheuristics, and machine learning. In machine learning, two main types of learning can be outlined: supervised and unsupervised. The former considers both input and output datasets in the learning algorithm, while the latter is mainly used in classification algorithms that do not require feedback and thus is less common in control problems. A well-known supervised classification algorithm is Classification and Regression Trees (CART) [19] or their extension, Random Forests [20]. CART can capture complex and nonlinear relations and can also be very helpful in detecting critical variables, in addition to other advantages in the field of automatic control [21]. Therefore, the CART algorithm is a well established method and it is well suited to the problem considered in this paper, since choosing which switches to commute is equivalent to a classification problem.

In this paper, the CART algorithm is used as a mean to directly obtain the results of the optimization problem from the system state information with no need to execute online traditional solvers. For this, the information obtained by solving the optimization problem offline along a full grid cycle [12,13], which is previously codified (the input–output dataset), is used to train several CARTs. These CARTs will associate the codified system state (input) with the corresponding codified optimal modulation (output). Afterwards, the resulting trees are implemented online using the same codification for the inputs and decodification for the outputs. In addition, by considering several operating points simultaneously in the training datasets, a more general solution is expected, which will overcome the limitations of using look-up tables which are valid for particular operating points. To increase the generality of the resulting trees, the use of quantitative information is avoided in the tree training. That is why raw input data is codified into qualitative variables, that provide enough information of the system state, before feeding the CART algorithm. In the same way, the output of the tree is a coded number with qualitative information about how to implement the modulation in each sampling time (it will be called modulation policy in the following).

Simulation and experimental results are presented showing the validity of the method for different working conditions compared to existing approaches.

In particular, the novelties of this paper compared to [13] are the following:

- The proposed method avoids the use of LUTs to obtain the modulation criteria and uses the decision trees obtained from the CART training. Such decision trees are less resource-demanding than LUTs.
- Several operating points are considered in the training datasets, achieving a general solution, as will be shown later in simulations and experiments.
- Additional performance features are introduced into the cost function to achieve best steady-state performance.
- A comparison by simulation with several approaches: LUT [13], FCS-MPC [11], OICM–MICM [10], and the original SVBA [7].
- Experimental comparison with the enhanced SVBA approach with balancing capabilities [8].

This approach, which combines offline optimization and Machine Learning to perform online modulation could be consider for other types of topologies. This could include MMC for HVDC, as the large number of submodules does not pose a problem for offline optimization resolution. Furthermore, additional objectives could be included in the cost function depending on the target application.

The outline of the paper is given as follows:

- Section 2: A description of the five-level DCC as a grid-connected rectifier is presented along with its mathematical model. Moreover, the strategy for controlling the currents and dc-link voltage of the converter that is later used in simulation and experiments is also introduced, highlighting the role of the modulation process on which this paper focuses on.
- Section 3: The modulation stage as an optimization problem is formulated and discussed in further detail as in [13]. Some improvements are presented. It is also shown that this optimization is carried out offline and, thus, the optimization execution time is not a problem for online modulation operation, but the need of an alternative for online implementation is demonstrated.

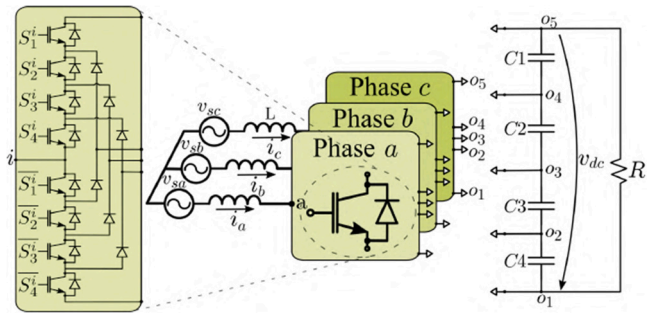


Fig. 1. Circuit of a three-phase five-level diode-clamped converter.

- Section 4: The pre- and post-processing stages for the optimization results as well as training and online implementation of the decision trees are presented.
- Section 5: The comparison in simulations in MATLAB-Simulink® with alternative control approaches is shown and discussed.
- Section 6: Experimental results are depicted and compared with other control techniques.
- Section 7: Some conclusions are drawn.

2. System model and control scheme

2.1. System model

This paper focuses on a three-phase, five-level DCC connected to the grid in AC/DC configuration. The diagram of this converter is shown in Fig. 1. The system is composed of a filter of inductance L, three phases with 8 switches and 6 diodes each, a dc-link (v_{dc}) composed of 4 capacitors — whose voltages are v_{c1}, v_{c2}, v_{c3} and v_{c4}, respectively –, and a resistor R connected to the dc-link to emulate the consumption of power. Phase currents and voltages are denoted as i_i and v_{si} for $i = a, b, c$, respectively. The output voltage of each phase i is measured from point i to dc-link middle point o_3. The switching states of each phase (f_{ij}) will be equal to 1 if phase i is connected to level j and 0 otherwise, for $i = a, b, c$ and $j = 1, \dots, 5$. For the dc-link capacitor voltage unbalance, variables v_{d1}, v_{d2}, v_{d3} are defined using the following system of equations

$$\begin{bmatrix} v_{d1} \\ v_{d2} \\ v_{d3} \\ v_{dc} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_{c1} \\ v_{c2} \\ v_{c3} \\ v_{c4} \end{bmatrix}. \tag{1}$$

The dynamical model of the system in $\alpha\beta$ can be retrieved by applying the Kirchhoff's laws (KL) to Fig. 1 and the Clarke transformation, resulting as follows

$$L\frac{d}{dt}\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} - \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}, \tag{2}$$

where parasitic resistors are obviated for the current dynamics and

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = (v_{c1}(t) + v_{c2}(t)) \begin{bmatrix} f_{\alpha 5} \\ f_{\beta 5} \end{bmatrix} + v_{c2}(t) \begin{bmatrix} f_{\alpha 4} \\ f_{\beta 4} \end{bmatrix} - v_{c3}(t) \begin{bmatrix} f_{\alpha 2} \\ f_{\beta 2} \end{bmatrix} - (v_{c3}(t) + v_{c4}(t)) \begin{bmatrix} f_{\alpha 1} \\ f_{\beta 1} \end{bmatrix}, \tag{3}$$

for the output voltage. System sampling and switching frequencies (for PWM implementation) have the same value f_s. By assuming that its corresponding period ($T = 1/f_s$) is sufficiently small, the switching function f_{ij} can be averaged by using the duty ratio d_{ij} instead, which expresses the fraction of the sampling interval in which phase $i = a, b, c$ is connected to point o_j. With this, the resulting model contains only continuous signals, and the constraints $\sum_{j=1}^{5} d_{ij} = 1$ for $i = a, b, c$ must be fulfilled. This averaging stage can be applied to Eq. (3), obtaining the duty ratios d_{kj} for $k = \alpha, \beta$. Note that one degree of freedom is made explicit in the $d_{\alpha\beta,j} \rightarrow d_{abc,j}$ transformation and will be used in the same way as [22] and will be explained later.

Assuming that the capacitors are equalized $v_{c1} = v_{c2} = v_{c3} = v_{c4} = v_{dc}/4$, variable v_k can be normalized and averaged as u_k:

$$u_k \doteq \frac{v_k}{\frac{v_{dc}}{4}} = 2d_{k5} + d_{k4} - d_{k2} - 2d_{k1}. \quad k = \alpha, \beta. \tag{4}$$

Using u_k for $k = \alpha, \beta$ as control signals, a standard current controller can be used to compute their values in such a way that the phase currents $\{i_\alpha, i_\beta\}$ track their references $\{i_\alpha{}^*, i_\beta{}^*\}$.

Once u_α, u_β are known, their transformation to abc leaves open one degree of freedom. This value will be called x and is limited by the definition of duty ratio, Eq. (4) and the modulation boundaries $u_i \in [-2, 2]$ for $i = a, b, c$, yielding an upper and lower bound for x. In summary,

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = T_{[\alpha\beta\rightarrow abc]} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x \tag{5}$$

$$u_i = 2d_{i5} + d_{i4} - d_{i2} - 2d_{i1}\ ; \quad u_i \in [-2, -1, 0, 1, 2] \tag{6}$$

$$\sum_{j=1}^{5} d_{ij} = 1 \quad i = a, b, c\ , \tag{7}$$

where $T_{[\alpha\beta\rightarrow abc]}$ is the inverse Clark transformation. The optimization problem that will be formulated in Section 3 will use this degree of freedom, x, and it will also consider constraints (6) and (7).

The dynamics of v_{d1}, v_{d2}, v_{d3} (1) are obtained from the capacitor dynamics and the definition of duty ratios as

$$C\frac{dv_{d1}}{dt} = \sum_{i=a,b,c} (-d_{i4} i_i) \tag{8}$$

$$C\frac{dv_{d2}}{dt} = \sum_{i=a,b,c} (-(d_{i1} + d_{i5}) i_i) \tag{9}$$

$$C\frac{dv_{d3}}{dt} = \sum_{i=a,b,c} (-d_{i2} i_i), \tag{10}$$

where terms $d_{ij}\, i_i$ refer to the averaged current of phase i that goes into node o_j.

Note that this formulation relates the duty ratio values with the current control fulfillment ($i_{abc} \rightarrow i^*_{abc}$) in (6), and with the capacitor balance issue in (8)–(10).

2.2. Control strategy

The main blocks of the control scheme used in this paper are shown in Fig. 2. Notice that this work focuses on the modulation block and how it deals with the capacitor voltage problem based on an optimization process and machine learning, as it will be detailed in Sections 3 and 4 (Fig. 4 shows the proposed real-time implementation of the modulation block).

A conventional two-block control scheme based on dc-link voltage and current controllers is used. However, alternative approaches based on averaged modeling, for example, could be used [23–25], as they can provide the modulation block with the required inputs. This fact also illustrates the flexibility of the proposed method.

The remainder of this section is devoted to briefly introduce the dc-link and current controller that are later used in simulation and experiments. It is worth mentioning that this is not the aim of the paper and, therefore, they are included briefly for the sake of completeness.

In grid-connected power converters, it is common to implement two cascaded controllers: dc-link voltage regulation and current reference tracking [26]. In this way, the dc-link voltage controller defines the references for the current controller. The former outputs the desired active power (p^*) in such a way that the dc-voltage reaches its reference

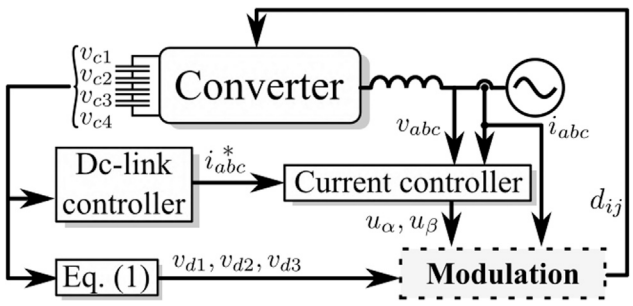


Fig. 2. Control scheme of the proposal. Notice that the paper focuses on the block "Modulation". For blocks "Dc-link controller" and "Current controller" different approaches can be used.

at steady-state. To achieve this, a simple PI controller is commonly employed [27]:

$$p^* = k_p^{dc}\left({v_{dc}^*}^2 - v_{dc}^2\right) + k_i^{dc}\int_0^t \left({v_{dc}^*}^2 - v_{dc}^2\right) d\tau \tag{11}$$

where k_p^{dc} and k_i^{dc} are the control parameters. Once the active power reference is known and the reactive power reference is given externally (q^*), the current references might be extracted as follows:

$$\begin{bmatrix} i_\alpha^* \\ i_\beta^* \end{bmatrix} = \frac{1}{v_{s\alpha}^2 + v_{s\beta}^2}\begin{bmatrix} v_{s\alpha} & -v_{s\beta} \\ v_{s\beta} & v_{s\alpha} \end{bmatrix}\begin{bmatrix} p^* \\ q^* \end{bmatrix}. \tag{12}$$

These references input the current controller, where the well-known non-ideal proportional-resonant (PR) controller [28] is used in this paper. The formulation for such controller is

$$G_{\mathrm{PR}}^{\omega_r}(s) = k_p + \frac{2K_r\omega_c s}{s^2 + 2\omega_c s + \omega_r^2}$$
$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \frac{4}{v_{dc}}\left(-G_{\mathrm{PR}}^{\omega_g}\left(\begin{bmatrix} i_\alpha^* - i_\alpha \\ i_\beta^* - i_\beta \end{bmatrix}\right) + \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix}\right), \tag{13}$$

where ω_r is the resonant frequency – tuned to be equal to the grid one (ω_g) –, ω_c is the cut-off frequency, and k_p and K_r are the control parameters. Thus, the values of u_α and u_β are obtained and then transformed to abc (5) once variable x is chosen. Capacitor voltage balance highly depends of this value, and still some degrees of freedom remain as the ones that are present in Space Vector Modulation. This stage is not trivial in five-level converters, above all if voltage balance is required (see, for example [7,8] where this problem is addressed with SVM related methods). This paper presents a different approach based on mixed-integer optimization for the computation of duty ratios and machine learning for the achievement of simple algorithms that are implementable online.

In the following, variables η_a, η_b and η_c are used as the corresponding values for u_a, u_b and u_c when x is chosen to be zero in Eq. (5):

$$\begin{bmatrix} \eta_a \\ \eta_b \\ \eta_c \end{bmatrix} \doteq T_{[\alpha\beta\to abc]}\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix}. \tag{14}$$

3. Modulation as an optimization problem

This section is devoted to present the modulation stage (shown in Fig. 2) as an optimization problem. Thus, it begins recalling the optimization procedure described in [12] and implemented in [13] through look-up tables, and several improvements in the formulation are made with respect to them. Therefore, a linear, mixed-integer optimization problem is considered, which obtains the duty ratios d_{ij} that accomplish u_α and u_β at the same time that other objectives, such as capacitor voltage balance and reduced number of commutations, are pursued. In contrast to [13], this paper resorts to machine learning to obtain decision trees that are used online to implement the results of the optimization approach. With this, an improvement in the robustness is achieved as it will be shown in Sections 5 and 6.

3.1. Optimization problem

This section presents the formulation of the linear mixed-integer optimization problem that could be solved at every sampling instant to determine which duty ratios d_{ij} for each phase $i = a, b, c$ and level $j = 1, \ldots, 5$ are chosen to be different from zero, i.e. which levels are used.

The following objectives are considered as constrains for the optimization problem [22]:

- The sum of the duty ratios for each phase i has to be 1 (7).
- The output of the power controller (u_α, u_β) (13) has to be accurately modulated in abc taking into account (4):

$$u_i = 2d_{i5} + d_{i4} - d_{i2} - 2d_{i1}. \tag{15}$$

- The balancing error signals cannot increase:

$$\mathrm{sign}(v_{dp})\frac{dv_{dp}}{dt} \le 0, \quad p = 1, 2, 3. \tag{16}$$

Considering (8)–(10), these constraints can be related to the duty ratios. Note that strict constraints in the sign of $\frac{dv_{dp}}{dt}$ for $p = 1, 2, 3$ might yield an increase in the number of commutations, that is why these constraints are softened to a non-strict inequality.

The desired behavior is prioritized by descending order of precedence:

1. Reduce as much as possible the number of commutations in order to reduce the switching losses. For this, the lowest amount of non-zero d_{ij} should be prioritized.
2. Large jumps within a phase must take place as little as possible. Large jumps are the sequential use of non-consecutive levels, and they can be expressed mathematically for a generic phase i as:

$$d_{ij_1} \neq 0;\ d_{ij_2} = 0;\ d_{ij_3} \neq 0 \text{ when } j_1 < j_2 < j_3$$
$$\text{for } \{j_1, j_2, j_3\} \in \{1, 2, 3, 4, 5\}. \tag{17}$$

Large jumps increase the current ripple and compromise the voltage limits of the devices, and thus they should be penalized.
3. Since the previous objectives are expressed in the cost function as integer numbers, it is likely that several points achieve the same score. In these cases, the points that fulfill the strict inequality in (8)–(10) should be prioritized over others, which is a point not considered in [12,13]. Furthermore, the signal error derivative magnitude may also be taken into account for the same purpose. These considerations are gathered in a new term 2 of the cost function as is shown in Eq. (18).

From [13] and adding term 2, the cost function results in

$$f_{\mathrm{cost}} = \underbrace{\sum_{i=a,b,c}\left(\sum_{j=1}^{5} s_{ij} + \sum_{m=1}^{6} p_{im}q_{im}\right)}_{\text{term 1}} + \underbrace{\sum_{p=1}^{3}\left(\alpha h_p - \alpha_1 \mathrm{sign}(v_{dp})\frac{dv_{dp}}{dt}\right)}_{\text{term 2}}. \tag{18}$$

A brief explanation of these variables and their associated constraints are given in the following.

- Integer variables $s_{ij} \ge 0$ for $i = a, b, c$ and $j = 1, \ldots, 5$ are greater than zero in the cases when level j in phase i is being used for the current solution, i.e. if duty ratio $d_{ij} \neq 0$, and equal to zero in the opposite case. To fulfill such purpose, the following linear constraint is defined:

$$s_{ij} - d_{ij} \ge 0 \text{ for } i = a, b, c;\ j = 1, \ldots, 5. \tag{19}$$

- Integer variables $p_{im} \in \{0,1\}$ and $q_{im} \in \{1,2,3\}$ are used to penalize large jumps in the solution candidate considered for each phase i. Large jumps are classified using index $m = 1,\ldots,6$ for the 6 types of large jumps: levels 1–3, 2–4, 3–5, 1–4, 2–5, and 1–5. Auxiliary variable p_{im} is introduced in order to register whether the large jumps are taking place ($p_{im} = 1$) or not ($p_{im} = 0$). A cost $q_{im} \in \{1,2,3\}$ is used in (18) that expresses the amount of non-used intermediate levels. For example, large jump 1–4 has two intermediate levels and two non-zero duty ratios – $s_{i1} = 1, s_{i4} = 1$ – then $q_{i4} = 2$. Besides, the detection stage, i.e. determination of p_{im}, for phase i must search for the occurrence of $d_{ij} \neq 0$ for $j = j_{m1}$ and $j = j_{m2}$ and $d_{ij} = 0$ for $j_{m1} < j < j_{m2}$. This can be considered by means of the following constraints.

$$s_{ij_{m1}} + s_{ij_{m2}} - r_{im} \leq 1; \quad 0 \leq r_{im} \leq 1 \tag{20}$$

$$r_{im} - p_{im} - \sum_{j=j_{m1}+1}^{j_{m2}-1} s_{ij} \leq 0; \quad 0 \leq p_{im} \leq 1, \tag{21}$$

  where $r_{im} = 1$ if the outer levels of the large jump candidate are used ($d_{j_{m1}} \neq 0$ and $d_{j_{m2}} \neq 0$).
- Integer variables $h_p \geq 0$ for $p = 1,2,3$ indicate whether $\frac{dv_{dp}}{dt}$ fulfills the strict inequality $\frac{dv_{dp}}{dt}\text{sign}(v_{dp}) < 0$ ($h_p \leq 0$) or not ($h_p > 0$). Therefore, the following constraint is added:

$$h_p - \frac{dv_{dp}}{dt}\text{sign}(v_{dp}) \geq 0, \quad \text{for } p = 1,2,3 \tag{22}$$

- Weighting factors α_1 and α satisfy $\alpha_1 \ll \alpha \ll 1$ in such a way that term 2 will only be relevant for candidates with the same value in term 1, prioritizing, firstly, the fulfillment of a larger amount of strict inequalities, and secondly, the voltage balance speed.

In summary, a linear, mixed-integer optimization problem is formulated. As it can be seen, such a problem has to be solved every sampling time, which may not be feasible for typical digital signal processors. Therefore, it is proposed to solve the problem offline and resort to Machine Learning to come up with a reduced procedure with an easy and fast implementation, as it will be exposed in Section 4.

3.2. Solution of the optimization problem

The optimization problem has to be solved each sampling period, resulting in the duty ratio values. However, the previous problem can hardly be solved at usual switching frequencies (around 5–10 kHz). For this, [12] proposes to solve it offline for N samples over a grid period under nominal operating conditions, recalling these results during online operation. Given the fact that there are 8 combinations of $\text{sign}(v_{dp})$, eight tables of results are obtained. Paper [13] accomplishes such implementation by resorting to look-up tables that used the angle of $u_{\alpha\beta}$ vector as index variable. Although this implementation was validated experimentally even with slight variations of the operating point, the LUT results were designed for specific operating conditions. Consequently, the same LUT will not necessarily work under different operating conditions, as will be shown later in experiments. This paper proposes to use the optimization results to come up with a solution that works in a wider range of operating points. For this, and contrary to [13], in this paper several operating points are considered, and the results of the optimization problem are stored in datasets.

It is shown in [12,13] that the solutions for every switching instant usually have the following characteristics:

- One phase (i_{fix}) fixed to one level (j_{fix}), i.e. it has one duty ratio equal to 1, and the remaining ones equal to zero.

$$d_{i_{\text{fix}}j_{\text{fix}}} = 1; \; d_{i_{\text{fix}}j} = 0 \quad \forall j \neq j_{\text{fix}} \in \{1,2,3,4,5\}$$

- Two phases (i_1 and i_2) that commute between two levels only (j_{11}–j_{12} and j_{21}–j_{22} respectively), i.e. they have two duty ratios different from zero, and the remaining ones equal to zero.

$$d_{i_1j_{11}} \neq 0; \; d_{i_1j_{12}} \neq 0; \; d_{i_1j} = 0 \quad \forall j \neq j_{11}, j_{12} \in \{1,2,3,4,5\}$$
$$d_{i_2j_{21}} \neq 0; \; d_{i_2j_{22}} \neq 0; \; d_{i_2j} = 0 \quad \forall j \neq j_{21}, j_{22} \in \{1,2,3,4,5\}$$

The idea is to use this knowledge about the solutions to the optimization problem to propose a codification procedure for the inputs and outputs of the CARTs. Thus, raw input and output data are transformed into qualitative integer variables before feeding the CART algorithm, as will be shown later. The data are arranged into eight datasets, one for each $\text{sign}(v_{dp})$ combination, obtained by simulating several operating conditions, and used to train a CART algorithm with the coded result being the output. Section 4 will provide more information on the coding of the inputs and results, as well as the training process. The resulting CARTs can be easily implemented online, and given the generality of the solutions of the CART learning algorithm and the codification of input and output data, a wide range of operating points can be covered, as will be shown in the simulation and experimental sections.

4. CART based modulation

The aim of this section is to depict how the previous optimal modulation can be implemented online without requiring to solve the optimization problem at every sampling time. For this, CARTs are trained with a wide variety of system states and the corresponding optimal modulation policies (solutions of the optimization problems as shown in Section 3.2). The resulting CARTs are implemented for online operation. Note that the optimization problem is solved for every combination of signs of v_{d1}, v_{d2}, v_{d3} as this factor imposes a different constraint on the optimization problem. Therefore, eight CARTs have to be obtained.

For this objective, several issues need to be tackled and elaborated in the following sections: (1) Codification of system state and resultant modulation policy into integer variables as CART inputs and outputs, respectively; (2) CART training; (3) Robust CART implementation for online modulation.

4.1. CART inputs and outputs codification

This section presents a codification procedure for the system state as well as modulation policies into integer variables. This codification procedure will later be used for both the training and the online implementation. As a result, each CART produces a coded output from any possible input data. For this, instead of using raw data from the continuous variables i_a, i_b, i_c, u_α and u_β, the input data should represent qualitative information of the system variables through discretized integer variables, improving CART training and achieving generality at the same time. Therefore, it is critical to select how to compute these integer variables (from i_a, i_b, i_c, u_α and u_β) and how to code the output.

4.1.1. CART inputs

In the following, the computation of the 22 categorized integer variables that are later used as inputs in the CART, either in the training and online implementation, is presented. For this, qualitative information is extracted from the optimization results and this information is fed into the CART generation algorithm. The qualitative information for each sampling time was chosen carefully using expert knowledge and the analysis of the optimization results for several test cases, such as:

- The sign of the currents i_a, i_b, i_c: -1 for negative values, and 1 for positive values. (3 variables)
- An integer number for each phase i (level_i) that indicates between which two levels the associated voltage reference η_i (14) is found. The code for this number is shown in Table 1. (3 variables)

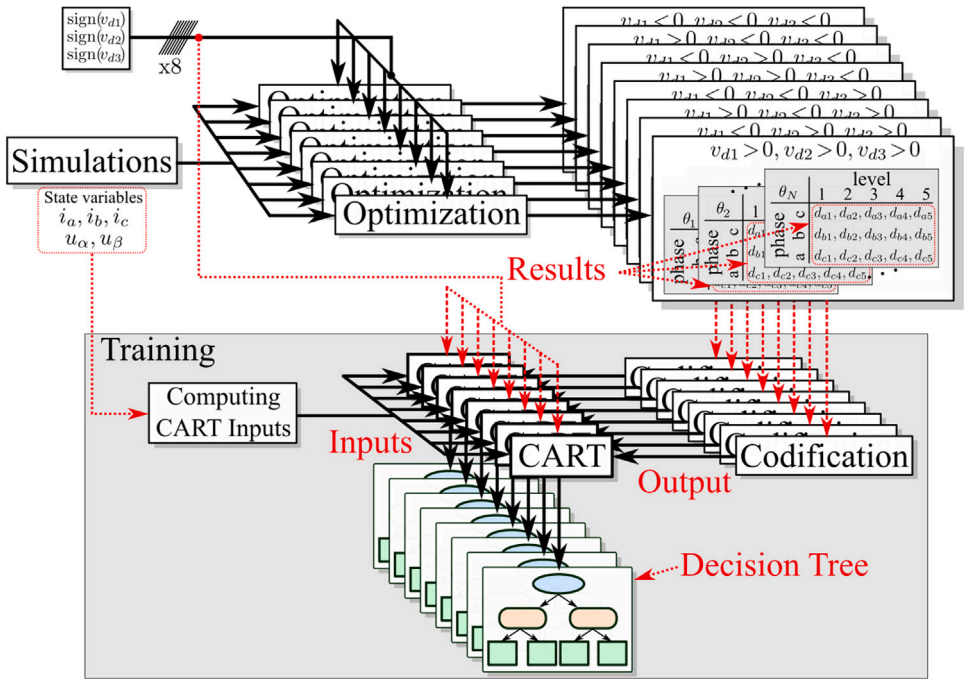


Fig. 3. Flowchart of carried out steps to obtain the eight decision trees based on the optimization problem.

Table 1
Integer number to indicate associated levels for η_i.

level$_i$	η_i	level$_i$	η_i
1	[−2,-1]	3	(0,1]
2	(−1,0]	4	(1,2)

Table 2
Integer number to indicate current order.

r	Current order	r	Current order
1	$i_a \geq i_b \geq i_c$	4	$i_c \geq i_b \geq i_a$
2	$i_a \geq i_c \geq i_b$	5	$i_b \geq i_c \geq i_a$
3	$i_c \geq i_a \geq i_b$	6	$i_b \geq i_a \geq i_c$

- A code (r) that indicates the order relation among i_a, i_b and i_c. This code is shown in Table 2. For example, when $i_a \geq i_b \geq i_c$ this code is equal to 1, when $i_a \geq i_c \geq i_b$ the code is equal to 2, and so on. (1 variable)
- Variables y_{ij} for $i = a, b, c$ and $j = 1, \ldots, 5$ are set to 1 when there exists a value of x that satisfies $x_{\min} \leq x \leq x_{\max}$ such that the corresponding value of $\eta_i + x$ is equal to j. This means that for the current sample data it is possible to consider a value of the homopolar component such that $u_i \in [-2, 2]$ is equal to level j, as it is shown in (5)–(7). Therefore, $y_{ij} = 1$ if $x_{\min} \leq j - 3 - \eta_i \leq x_{\max}$, and $y_{ij} = 0$ otherwise. (15 variables)

The previous 22 variables are computed for every sample from i_a, i_b, i_c, u_α and u_β, and will be used as representative of the current state of the system for the CART.

4.1.2. CART outputs

Given one sample of the system, the optimization problem can be solved. It is assumed that the optimization problem result dictates one phase to be fixed at one level, while the other two phases commute between two different levels. Therefore, a code that captures this information is defined by using three numbers a_1, a_2, a_3:

- The first number a_1 indicates the pair of values i and j such that $d_{ij} = 1$, i.e. which phase i is fixed at which level j. Table 3 shows this codification (15 possible values)

Table 3
Integer number a_1 to indicate which d_{ij} equals one.

a_1	(i,j)	a_1	(i,j)	a_1	(i,j)	a_1	(i,j)	a_1	(i,j)
1	(*a*,1)	2	(*a*,2)	3	(*a*,3)	4	(*a*,4)	5	(*a*,5)
6	(*b*,1)	7	(*b*,2)	8	(*b*,3)	9	(*b*,4)	10	(*b*,5)
11	(*c*,1)	12	(*c*,2)	13	(*c*,3)	14	(*c*,4)	15	(*c*,5)

Table 4
Integer number a_2/a_3 to indicate pair of used levels.

Code	Levels	Code	Levels	Code	Levels
1	1 – 2	5	1 – 3	8	1 – 4
2	2 – 3	6	2 – 4	9	2 – 5
3	3 – 4	7	3 – 5	10	1 – 5
4	4 – 5				

- Numbers a_2 and a_3 represent the pair of levels that the remaining phases use. The correspondence of a_2 and a_3 with the phases follows alphabetical order: for example, if the phase used for code a_1 is phase b, code a_2 corresponds to phase a and code a_3 to phase c. The pair of levels are coded by enumeration as shown in Table 4. (10 possible values each)

These values can be combined into a single variable with $15 \times 10 \times 10 = 1500$ possible values. This variable is used as the output for the CART algorithm and it is coded as follows

$$\text{Output code} : (a_1 - 1) \cdot 10^2 + (a_2 - 1) \cdot 10 + (a_3 - 1). \tag{23}$$

4.2. CART training

In this section, the CART inputs and outputs, as described in Section 4.1, are gathered into datasets to perform the training of the CARTs. This information comes from offline simulations where N samples over a grid period at steady state conditions for several operating points are obtained to feed the optimization problem. Both the system state and the optimization result for every sample are stored to form the training dataset. This training operation has to be computed for every possible combination of signs of v_{d1}, v_{d2}, v_{d3}, obtaining as a result 8

Table 5
CART training parameters.

Parameter	Value	Parameter	Value
Samples per operation point and grid period N	100	Optimization weighting factor α_1	0.01
Optimization weighting factor α	0.001	CART complexity parameter cp	0.002
Training dataset coverage	>85%	Max. number of levels	11
Number of operating points (Table 6)	6	Integer input variables	22

Table 6
Operations points considered for the training data set.

No.	v_{dc} (V)	p (kW)	q (kVA)
1	800	10	0
2	800	0	10
3	800	0	−10
4	700	10	0
5	700	0	10
6	700	0	−10

decision trees. Fig. 3 shows the summary of the training stage for one operating point. In order to extend the operability of this approach, additional operating points are considered, simulated and added to the corresponding dataset. The training parameters and information on the resulting trees are included in Table 5, and the considered operating points are shown in Table 6.

The trees are generated using the *recursive partitioning and regression trees* technique from the *rpart* library for R language. The arguments of this function are: the input–output variables; the classification method set as "class"; the way the decisions are considered, in this case, through the complexity parameter cp, which determines that a split is not carried out if it does not decrease the overall lack of coverage by the cp value; and a vector with the cost values. The latter is a vector that contains one cost value for each input such that the inputs with increased cost would appear less frequently in the decision tree. The cp value is tuned in such a way that a minimum of 85% of coverage of the training data set is obtained.

The input cost values are justified and selected as:

- The sign of the currents and their order code (r) keep the same value for a large number of samples within one grid period, and thus they should be consulted less frequently: Cost value = 5
- Variables level$_i$ are the next considered as they change more frequently: Cost value = 2.5
- Variables y_{ij} are the easiest to classify due to their binary nature, while they also provide useful information about the feasible values of x: Cost value = 1.

In summary, there exist 22 variables to be used as input parameters and one coded integer result whose value could range from 0 to 1500. An example of a generated tree is depicted in Fig. A.14 in Appendix. The maximum number of levels for all trees is 11, and they are balanced with a similar percentage of coverage.

4.3. Implementation for online modulation

The computed decision trees can be easily implemented just by means of simple nested "if...else" statements, making it possible to be evaluated in real time in order to perform the modulation block within the control loop shown in Fig. 2. Thus, Fig. 4 shows the detailed real time implementation scheme for the modulation block, which consists of the following steps:

4.3.1. CART selection

During online operation, the decision tree that corresponds to the actual combination of sign(v_{dp}), $p = 1, \ldots, 3$, (see Eq. (1)) is selected, as shown in Fig. 4.

4.3.2. Computing CART inputs and evaluation

This CART is fed with the input data parameters computed from the actual currents i_{abc} and voltage commands $u_{\alpha\beta}$ (see Section 4.1.1). With these inputs, the "if...else" statements of the CART can be evaluated and, thus, the corresponding output code is obtained. (see Section 4.1.2).

4.3.3. Decodification

In order to obtain the values of the duty ratios d_{ij} from the output code, the following procedure is considered,

I. Variables a_1, a_2 and a_3 are obtained from the output code. To depict an example, let us consider an output code of 574 from Fig. A.14. Variables a_1, a_2, a_3 are extracted from the output code by means of simple integer division:

$$a_1 = \text{int}(\text{code}/10^2) + 1 = 6 \tag{24}$$

$$a_2 = \text{int}((\text{code} - (a_1 - 1) \cdot 10^2)/10) + 1 = 8 \tag{25}$$

$$a_3 = \text{int}((\text{code} - (a_1 - 1) \cdot 10^2 - (a_2 - 1) \cdot 10)) + 1 = 5 \tag{26}$$

II. Using (5)–(7) with u_α, u_β obtained from (13), and the information provided by a_1, the value of x is obtained.

III. Knowing x and the information provided by a_2 and a_3, the remaining d_{ij} are computed according to (5)–(7).

4.3.4. Duty cycle computation

To guarantee a robust implementation of the desired voltage vector, for example, when the u_α, u_β values cannot be represented using the levels indicated by a_1, a_2 and a_3, we propose the following procedure:

I. The result is decoded and the levels to be used for each phase are obtained according to a_1, a_2 and a_3: fixed phase i_{fix} at level l_{fix}; phases i_1 and i_2 fixed at j_{11}, j_{12} and j_{21}, j_{22}, respectively.

II. It is checked if the indicated fixed phase (i_{fix}) can be fixed at level (l_{fix}), that is:

$$x_{\min} \leq \underbrace{l_{\text{fix}} - 3 - \eta_{i_{\text{fix}}}}_{x_{\text{fix}}} \leq x_{\max}. \tag{27}$$

If so, then set $x = x_{\text{fix}}$, otherwise x is saturated to the closest value $x_{\min}$ or $x_{\max}$ such that one phase is still fixed at one level.

III. A similar procedure is carried out for phases i_1 and i_2. For each of them, they should be modulated using the indicated levels taking into account the previous x value. This is equal to check if:

$$j_{11} \leq \eta_{i_1} + 3 + x \leq j_{12} \tag{28}$$

$$j_{21} \leq \eta_{i_2} + 3 + x \leq j_{22}. \tag{29}$$

In the case this check is not fulfilled, the used levels are changed to the nearest levels. That is, the closest levels above and below of $(\eta_i + 3 + x)$. Once the two levels for each phase, for which the corresponding duty cycles are not zero, the computation of them is obvious using trivial and well-known weighting formulae.

Because of this procedure, the implemented duty ratios might not be the optimal values according to the formulated problem, which is the reason why this work is entitled pseudo-optimal modulation. Additional sources of error that could make the implemented result to diverge from the optimal one are:

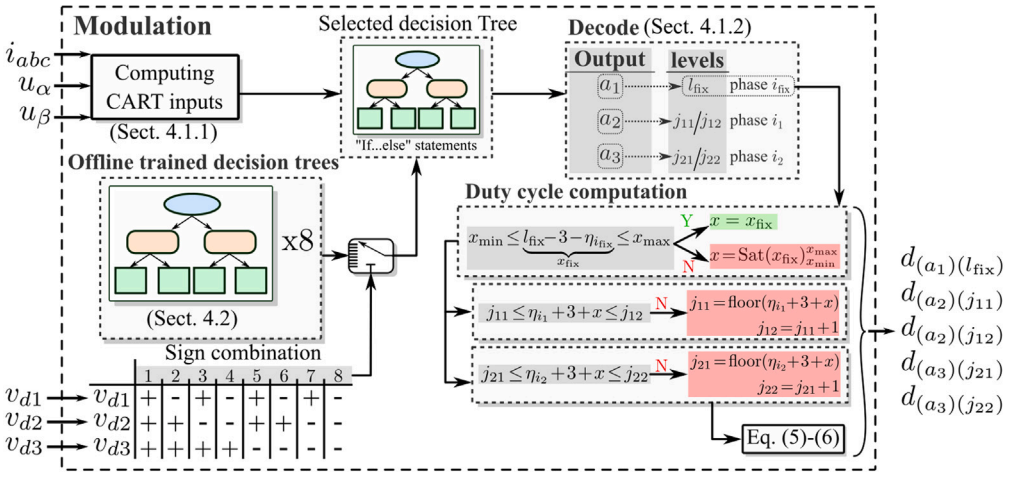


Fig. 4. Implementation scheme for real time modulation block implementation with robust modulation of u_α, u_β.

Table 7
Converter parameters.

Parameter	Value	Parameter	Value
Grid frequency f	50 Hz	Sampling frequency f_s	10 kHz
Grid voltage v_{sabc}	230 V_{RMS}	Switching frequency f_{sw}	10 kHz
Inductance L	2 mH	Prop. term PR controller k_p	5
Capacitance C	3300 μF	Resonant term PR controller K_r	50
dc-link load R	$[60, \infty]$ Ω	Prop. term dc-link controller k_p^{dc}	0.05
dc-link voltage reference v_{dc}^*	[700, 800] V	Integral term dc-link controller k_i^{dc}	1
Low-pass cut-off frequency (PR) ω_c	$2\pi 5$ rad/s	Resonant frequency (PR) ω_r	$2\pi 50$ rad/s

Table 8
THD values of the currents (Fig. 5(a)) and number of commutations over a grid period (Fig. 5(b)).

Approach		THD (%)	# Commt
CART		4.6	550
LUT		4.6	645
SVBA [7]		7.01	935
FCS-MPC [11]	10 kHz	13.42	210
	25 kHz	5.10	540
	50 kHz	2.54	1100
OICM [10]		5.40	1600
MICM [10]		4.2	1333

I. The measured variables, as well as the values of u_α and u_β, may not correspond exactly to any of the training data sets, and therefore the result is inferred from the training datasets.
II. The CART algorithm has a coverage error and not all cases used for the training are correctly matched in the output given by the decision trees.
III. There were some samples that did not fulfill the generic rule of one phase fixed to one level and the others using only two levels, thus those few cases are neglected with the current approach — less than 3% in any of the solved optimization problems.

Despite these sources of errors, it will be shown in the following sections that the results are satisfactory. Overall, the implementation that has been used in this paper is depicted in Figs. 2 and 4, where it can be seen that the contribution lies in the modulation and voltage balancing stage. In other words, the only requirement is to give the phase currents i_{abc} and the voltage commands $u_{\alpha\beta}$ to the pre-processing stage (apart from the signs of v_{d1}, v_{d2} and v_{d3} to select the appropriate decision tree), which can be derived from any control technique that relies on voltage vector modulation

4.4. Summary of CART based modulation and remarks

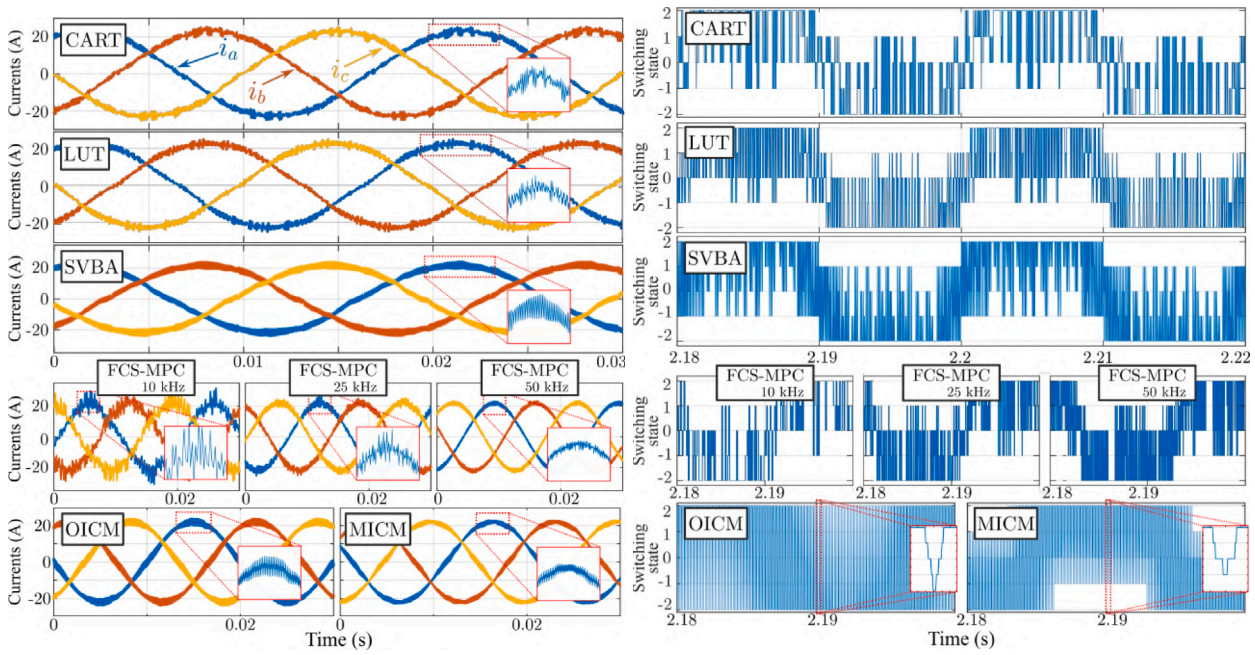
In summary for this approach, for each of the 8 possible combinations of v_{d1}, v_{d2}, v_{d3}, the following steps were carried out:

1. The optimization problem is solved offline under fixed operating conditions. Such optimization would determine which levels to use for every sampled instant, obtaining a table for those fixed operating conditions. Several operating conditions can be gathered in a database. (Section 4.1.2)
2. For each sampling instant in the database, the 22 integer CART inputs are obtained (from i_a, i_b, i_c, u_α and u_β), and the results, i.e. which levels should be used in the modulation, are codified. (Section 4.1)
3. The CART is trained offline with the previous information. As a result, a decision tree (nested if-else) is obtained. Such a tree yields a codified output when a string of inputs is given. (Section 4.2)

Once all 8 offline decision trees have been computed, the online implementation (Section 4.3) consists of only the execution of the decision tree considering the inputs obtained from the measurements, as shown in Fig. 4.

Remark 1. This approach proposes a modulation stage to solve the balancing capacitor voltages, but at the same time, the current control is always fulfilled and given maximum priority.

Remark 2. This approach is based on the model of the five-level DCC, as the optimization is based on the capacitor voltages balancing dynamics. However, other multilevel converters with different dynamics can be considered as long as the optimization problem is modified accordingly. Furthermore, given that the desired output of the current control is always prioritized and that the balancing is executed to modify the sign of the derivatives of v_{d1}, v_{d2}, v_{d3}, without paying attention to their magnitude, this approach fulfills the control objectives regardless of the



(a) Three phase currents. (b) Switching state of phase a.

Fig. 5. Simulation: Steady-state performance for the proposed approach (top), the LUT-based (second row), the SVBA (third row), the FCS-MPC for three different sampling and switching rates (fourth row), and the OICM and MICM (bottom). $v_{dc} = 800\,\text{V}$, $R = 60\,\Omega$, $q^r = 0$ kV A.

	p (kW)	q (kVA)	S (kVA)	PF
Test A	0	5	5	0
Test B	4	8	8.9	0.45
Test C	4	0	4	1
Test D	4	-8	8.9	0.45
Test E	0	-5	5	0

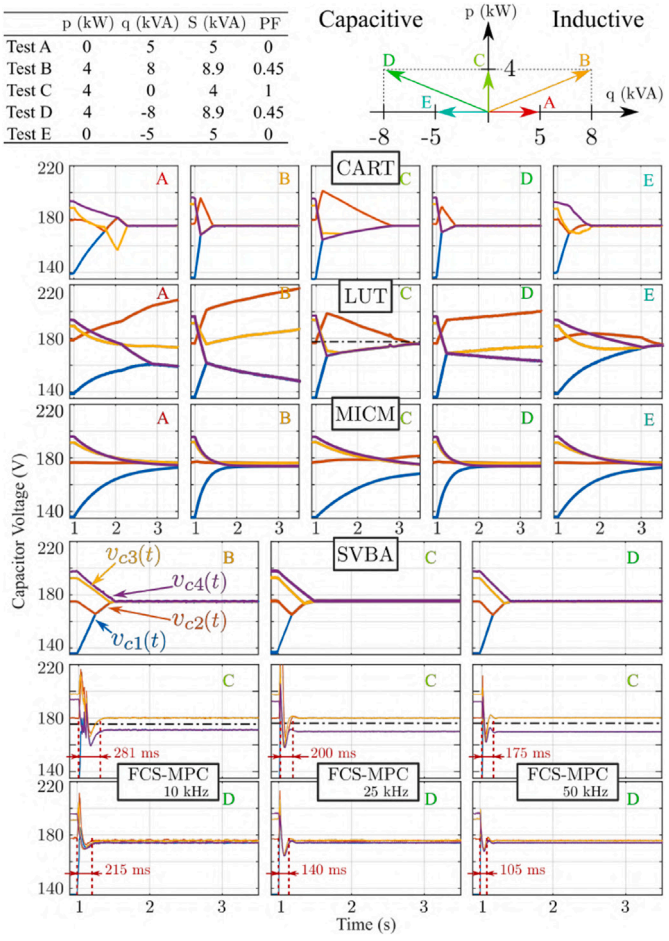


Fig. 6. Simulation: Capacitor voltages evolution starting from an unbalanced situation $v_{d1} = -40, v_{d2} = 60, v_{d3} = -5$ for all the considered approaches under different tests whose power references are shown at the top of the figure. $v_{dc} = 700\,\text{V}$.

system parameters, such as capacitance or phase inductance (within the boundaries of active and reactive power and modulation index).

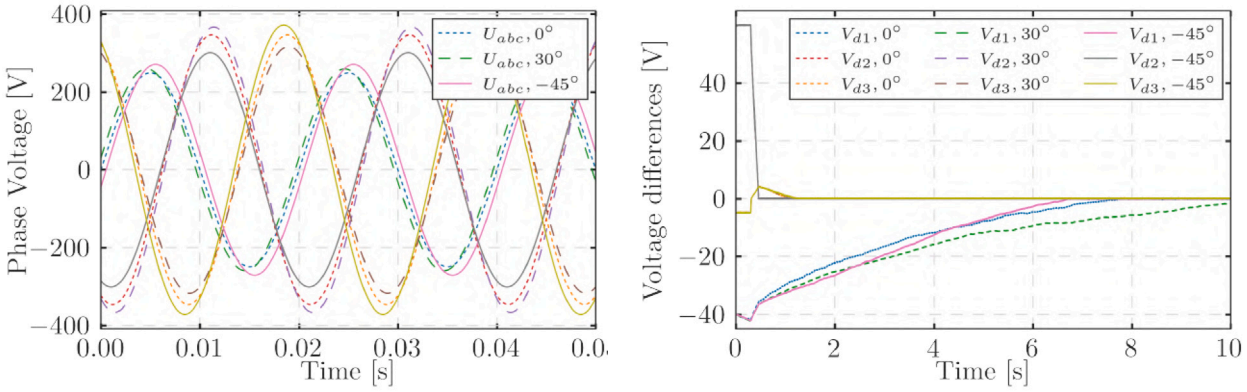
5. Simulation results

Before testing the proposed technique in a real environment, some in-depth simulations have been performed in MATLAB-Simulink®. The operation points of the offline simulations considered to generate the training datasets are the same shown in Table 6. For a proper comparison, 5 different modulation methods, with balancing capabilities, for 5-level DCC converters are considered apart from the proposed one (CART):

- A space-vector based algorithm [7] (SVBA).
- Two integrated control and modulation methods proposed in [10] (OICM and MICM).
- A model-predictive control based on finite control set (FCS-MPC) [11].
- The original proposal for the optimization problem implemented through a look-up table [13] (LUT).

Among these six controllers, LUT, CART and FCS-MPC are based on optimal formulations, whereas SVBA, OICM and MICM are based on conventional approaches that uses averaged models.

The LUT method is an alternative procedure to implement the results of the offline optimization based on interpolating them from a look-up table. The SVBA combines the small and large switching vectors in the space vector hexagon in such a way that balancing capabilities are achieved for any operating point. The FCS-MPC is based on selecting every sampling interval the switching state that yields the lower value of a weighted cost function that covers current reference tracking and voltage balance while minimizing the number of commutations. The cost-function weighting values of the FCS-MPC have been selected in such a way that the balancing error at steady state is lower than 10 V. Moreover, the FCS-MPC approach has been evaluated under three sampling frequencies: 10, 25 and 50 kHz. Lastly, OICM and MICM define a control law to compute the duty ratios for each phase and level in the control stage.



(a) Simulation: 3-phase grid unbalanced voltages.

(b) Simulation: Capacitor voltage error evolution.

Fig. 7. Simulation: (Left) The three cases of grid voltages with negative sequence component with phase-shifts: 0°, 30° and −40°. (Right) Resultant capacitor voltage differences for the proposed approach.

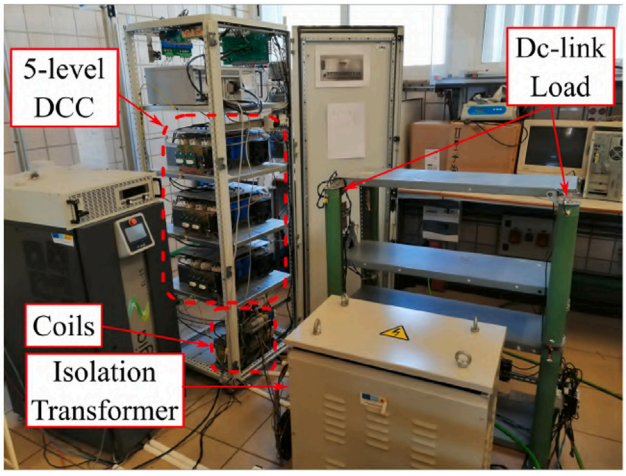


Fig. 8. Experimental setup used for the experiments.

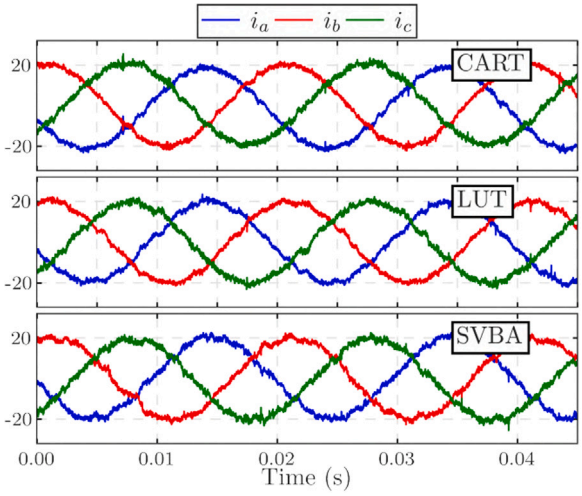


Fig. 9. Experiment: Three-phase currents in steady-state for CART, LUT [13] and SVBA [8]. $v_{dc} = 750\,\text{V}$, $R = 60\,\Omega$, $q^r = 0$ kV A.

Table 7 shows the circuit and controller parameters which will be used both in simulations and experiments. Table 6 shows the operating points considered in the training dataset (see Section 4.2 for details on the offline training process).

The phase currents in steady state for each of the modulation methods are depicted in Fig. 5(a) with a zoomed area of the switching ripple. The values of the current total harmonic distortion (THD) computed in simulation are shown in Table 8. The number of commutations is also shown for each method in Table 8 based on the switching state output depicted in Fig. 5(b).

5.1. Currents and commutations

The proposed method achieves a current distortion similar to that of the LUT-based method and smaller number of commutations, which is an expected outcome as both implement the same optimal formulation and the CART generalizes the result of the optimization. SVBA and OICM have larger THD values due to the fact that they tend to use larger switching vectors, increasing the effective phase-phase voltage and number of commutations. In contrast, MICM achieves better current distortion at the cost of yielding more than twice the number of commutations. FCS-MPC deserves special attention, as there is a trade-off between the THD value and the number of commutations according to the sampling frequency used. At 25 kHz, FCS-MPC achieves similar current distortion and number of commutations as CART, but requires an increase in the sampling frequency, which may not be feasible in certain scenarios.

5.2. Capacitor voltage balance

The capacitor voltage balancing capabilities are shown in Fig. 6 under several scenarios of power factor (depicted at the top of the figure). This figure shows how the capacitor voltage values evolve when the balance starts at $t = 1$ from an unbalanced voltage condition ($v_{d1} = -40, v_{d2} = 60, v_{d3} = -5$ for $t < 1$). In comparison with LUT, CART is able to achieve balancing under a wider range of operating points. The LUTs were generated considering only the conditions of Test C, therefore the balancing is not guaranteed for different conditions, as happens with tests A, B, and D. On the contrary, due to the extended training data set, the qualitative information used for input–output data and its generalization capability, the CART approach is capable of achieving voltage balance capabilities, thus achieving a more flexible implementation of the optimization problem originally computed. It should be mentioned that the power factors of tests B and D were not explicitly considered in the training dataset (Table 6), which indicates the generalization capabilities of the CART method.

In terms of capacitor voltage balancing of the remaining approaches, FCS-MPC exhibits the fastest balancing no matter the sampling frequency or the test considered — which is why only two tests for the

three sampling frequencies have been considered. However, no zero-steady-state error is achieved. Alternatively, SVBA shows improved balancing capabilities with the downsides of increased current distortion and number of commutations (see Table 8). For the sake of clarity, only the MICM approach is depicted, as its balancing capabilities are better than those of the OICM. Still, MICM has the slowest balance of the considered approaches without reaching zero-steady-state error. Note that the differences in the settling time between Tests B, C and D for SVBA and Test C and D for FCS-MPC have little effect on this comparison. Therefore, for the sake of clarity, Tests A, E in SVBA and Tests A, B, E for FCS-MPC are omitted here. Lastly, CART achieves balancing times similar to LUT, which are faster than MICM, but slower than SVBA or FCS-MPC. Nevertheless, this balancing times are not expected to have any effect during steady state conditions, only during transients, and thus it is mainly required that balanced conditions are reached.

5.3. Operation under grid unbalanced conditions

Despite the fact that CART training has not considered an unbalanced grid, the method still achieves balancing in the presence of negative sequence. This is because the current control is always fulfilled and the modulation only changes the derivatives of the capacitor voltage differences, no matter their magnitude. This is shown in Fig. 7 where capacitor voltage balancing capabilities are shown for different negative sequence conditions in the grid voltage. For the cases considered, a phase voltage reduction of up to 20% in one phase is given, and the proposed method should keep the capacitors balanced. Different phase shifts of the negative sequence are considered and shown in Fig. 7(a).

The resultant capacitor voltage differences are shown in Fig. 7(b). It can be seen that in all cases the voltage differences are driven to zero in a reasonable time. The behavior of the phase currents is omitted here because it is almost identical to the top graph in Fig. 5(a).

5.4. Summary of simulations

In summary, for these simulations, the proposed method achieves an acceptable balancing settling time, a good THD value, and a low number of commutations, even in scenarios that were not covered in the training dataset. Therefore, it performs better compared to the rest of the approaches considered.

6. Experimental results

In this section, several experiments have been carried out with three different modulations. The purpose of this comparison is to analyze the feasibility and improvement of the current proposal compared to the alternative implementation (LUT) and a more conventional strategy (SVBA). These modulations are:

1. CART: Proposed modulation with the same trees obtained in the simulation section (Section 5).
2. LUT: Alternative procedure presented in [13] for the implementation of the optimization results. It uses the same tables as the one obtained for the simulations.
3. SVBA: space vector approach with balancing capabilities extracted from [8] as Improved Vector Decomposition SVM, which modifies the vector sequence of [7] – used in simulation – to improve current distortion and reduce commutations.

The system used for the experiments is shown in Fig. 8, and the same decision trees and parameters (Table 7) used in the simulation section are considered.

Firstly, the phase currents obtained with the three approaches are shown in Fig. 9, and their corresponding THD value in Fig. 10 for $v_{dc} = 750\,\mathrm{V}$ and $R = 60\ \Omega$ at steady-state conditions and nominal

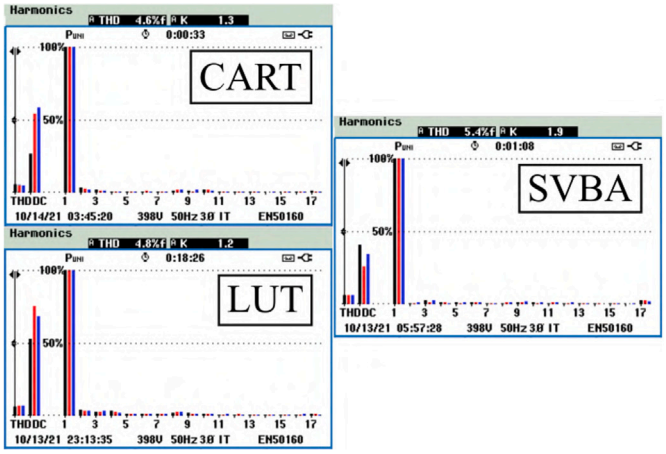


Fig. 10. THD values of the currents shown in Fig. 9.

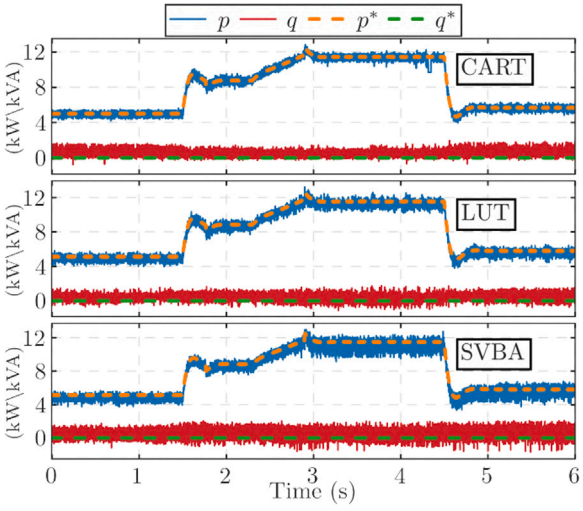


Fig. 11. Experiment: Test 1. Active and reactive power behavior of the considered three approaches (Table 9).

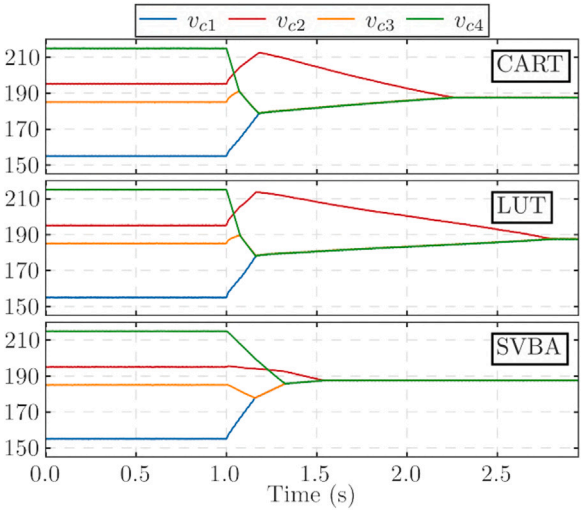


Fig. 12. Experiment: Test 2. Capacitor voltage balancing under unitary power factor conditions when starting from an unbalanced capacitor voltage condition ($v_{d1} = -40$, $v_{d2} = 60$, $v_{d3} = -30$). $v_{dc}^{*} = 750\,\mathrm{V}$.

Table 9
Load and dc-link voltage values used in Test1.

$t: 0 \rightarrow 1.5$		$t: 1.5 \rightarrow 2.2$		$t: 2.2 \rightarrow 2.9$		$t: 2.9 \rightarrow 4.4$		$t: 4.4 \rightarrow 5.5$	
v_{dc}^* (V)	R (Ω)	v_{dc}^* (V)	R (Ω)	v_{dc}^* (V)	R (Ω)	v_{dc}^* (V)	R (Ω)	v_{dc}^* (V)	R (Ω)
750	120	750	60	750 → 800	60	800	60	800	120

operating point. In spite of CART and LUT having different implementation approaches, both are based on the same optimization problem (except for term 2 in (18) which is exclusive of this paper, although it does not introduce noticeable differences in steady state). Therefore, they yield very similar THD values, despite the fact that the CART implementation is much more simple, among other advantages that will be exposed in the following. Alternatively, the SVBA approach improves the distortion from the simulations, but it is still larger than CART or LUT in experiments. This is due to the fact that SVBA resorts to small and large switching vectors within every sequence to allow the use of redundant ones, which increases the switching ripple.

To evaluate the behavior of the system under different conditions, three experimental tests are considered:

- Test 1: Dynamical test on modulation index and active power. In this test, the values of the dc-link voltage reference (v_{dc}^*) and the load (R) are modified during online operation. These values are shown in Table 9.
- Test 2: Capacitor voltage balancing under unitary power factor condition. In this test, the system starts in an unbalanced situation (similar to Test C in Fig. 6) with $v_{d1} = -40, v_{d2} = 60, v_{d3} = -30$ and the balancing is activated at $t = 1$. $v_{dc}^* = 750$ V and $R = 120\,\Omega$.
- Test 3: Dynamical test on reactive and active power. In this test, the reactive power reference (q^*) of the system is modified from zero to 5 kV A at $t = 1$, and to -5 kV A at $t = 4$. In parallel, the load is detached at $t = 2$ to seek zero power factor. The power conditions of this test is equivalent to those found in Test C, Test B, Test A and Test E in simulation (Fig. 6). $v_{dc}^* = 750$ V and $R = [120, \text{inf}]\,\Omega$.

The first test is devoted to show how the CART approach performs similarly to the LUT-based one when the operating conditions are those for which the LUT was designed, and were considered, among others, in the CART training dataset. Besides, the results when using SVBA are also depicted. For this, the active and reactive powers, together with their references, are depicted in Fig. 11. It can be seen that the changes in v_{dc} or R considered in Table 9 modify p^* accordingly. When comparing the three approaches, conclusions similar to those corresponding to Fig. 9 can be extracted. The three approaches behave satisfactorily, where CART and LUT present a similar ripple in the instantaneous active and reactive power, while SVBA presents a slightly larger ripple. The latter is consistent with the THD values shown in Fig. 10.

Test 2 is focused on the capacitor voltage balancing capabilities under unitary power factor conditions, i.e. the power factor under which the LUT was generated and were also included in the training dataset. As can be seen in Fig. 12, the capacitor voltage balancing takes place at $t = 1$, achieving a balanced condition in 1.2, 1.8 and 0.55 s for CART, LUT and SVBA, respectively, which is similar to the results obtained in simulations (test C in Fig. 6). Note that, this is a transient state and it is only shown to prove that the capacitor voltage differences are corrected in the three scenarios under nominal conditions, therefore the balancing speed is of little concern for the steady-state performance.

Lastly, Test 3 is intended to show the feasibility of the three modulation approaches under non-unitary power factor conditions. For this, the reactive power reference is modified from 0 to 5 kVA at $t = 1$, and to -5 kVA at $t = 4$. In parallel, the load starts at 120 Ω and is detached at $t = 2$ to emulate a power factor equal to zero. The results are shown in Fig. 13. As can be seen, both CART and SVBA keep the capacitors balanced under these changes, while LUT fails to achieve this objective, similarly to the simulation results (tests B, D, and E in Fig. 6). The main reason is that the LUTs were not obtained under such power factor conditions, and therefore capacitor voltage balancing is not guaranteed. In fact, the capacitor voltages start to diverge from $t = 1$ until the experiment has to be stopped to avoid capacitor overvoltage. In contrast, and as one of the main benefits of CART versus LUT, the CART approach is able to implement the

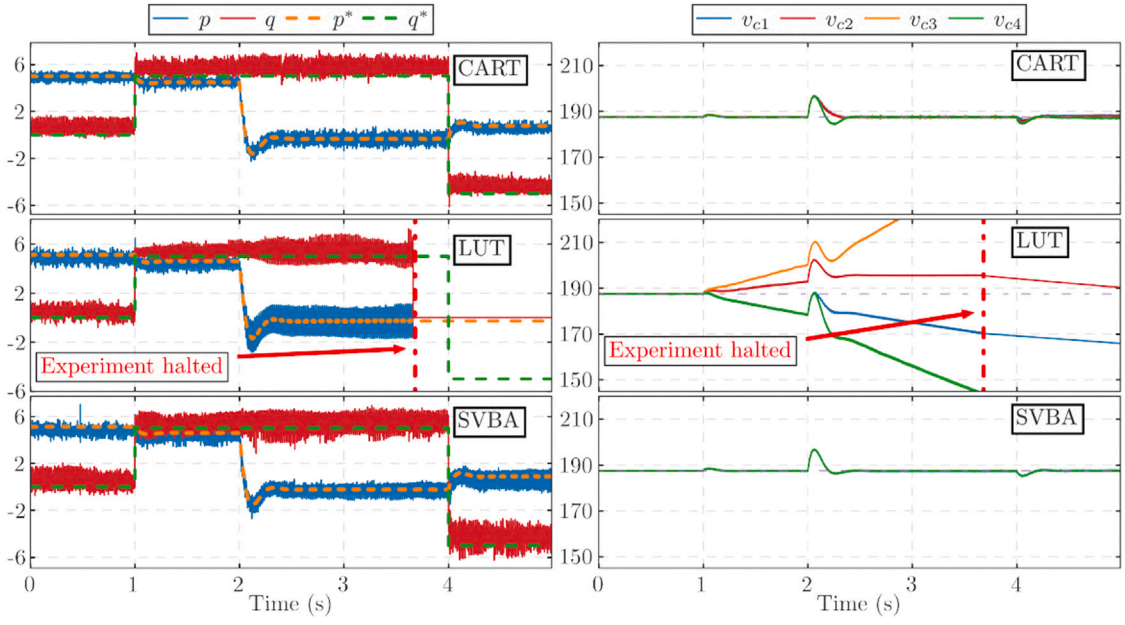


Fig. 13. Experiment: Test 3. (Left) Active and reactive power under changes in the reactive power ($q^* = [0, 5, -5]$ kVA at $t = 1$ and $t = 4$), and when the load is detached from the dc-link at $t = 2$. (Right) Capacitor voltages evolution under the same experiment.

Table 10
THD values of the currents (Fig. 9) and number of commutations over a grid period.

Approach	THD (%)	# Commt
CART	4.6	528
LUT [13]	4.8	610
SVBA [8]	5.4	752

optimization results and fulfills this objective under non-unitary power factor values. In fact, it is worth noting that the CART was trained only for the conditions shown in Table 6, where condition $q^* \neq 0$ and $p^* \neq 0$ was not included. However, in period $1 \leq t \leq 2$, this condition is present and the CART is able to maintain the capacitors balanced. Therefore, this approach has been able to consider other operating points different from those for which it has been trained, which indicates its robustness and feasibility for wider operating points.

The average number of commutations per grid period produced by each approach is measured in the period from $t = 0$ to $t = 1$ of Test 1 (Fig. 11), yielding the results shown in Table 10. The proposed approach stands out for its reduced THD value and the number of commutations. Compared to SVBA, the CART approach has a lower balancing speed, as shown in Fig. 12. However, let us remark that the balancing is a secondary objective that takes effect during a transient time. Once it is fulfilled, the performance indicators, such as THD and switching losses, are more critical.

7. Conclusions

This paper presents a new method to implement the modulation stage for 5-level DCCs based on a linear, mixed-integer optimization. Such a problem is solved offline and its solutions are learned using a classification and regression tree (CART) procedure. As a result, eight regression trees are obtained that can be easily implemented in low-resource control hardware by means of simple nested "if-else" statements. Furthermore, the resulting modulation algorithm achieves capacitor voltage balance under a wide range of operating conditions using three features: (1) consideration of several operating conditions in the training dataset; (2) the way the qualitative information is captured for the input–output training data; and (3) the generalization capabilities of CARTs. The resulting modulation scheme has been tested for several operating conditions, including some not covered in the training set, showing enhanced effectiveness compared to other approaches in both simulation and experiments. Additionally, random forests have become an increasingly popular extension of CARTs, exploiting their randomness to produce even better decision trees, and thus it may be advisable to extend this work in that direction. Other classifications schemes, such as C4.5, might offer additional benefits on the coverage and implementation while taking into account the implementation restrictions. In that case, no changes should be made in the control architecture, the controller implementation, or even the optimization problem formulation and solution. Furthermore, this approach could be easily extended to alternative topologies by redefining the optimization formulation.

CRediT authorship contribution statement

Pablo Montero-Robina: Study conception and design, Data collection, Analysis and interpretation of results, Writing – original draft. **Francisco Gordillo:** Study conception and design, Writing – original draft. **Fabio Gómez-Estern:** Study conception and design, Writing – original draft. **Federico Cuesta:** Analysis and interpretation of results, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

All authors reviewed the results and approved the final version of the manuscript.

Appendix

See Fig. A.14.

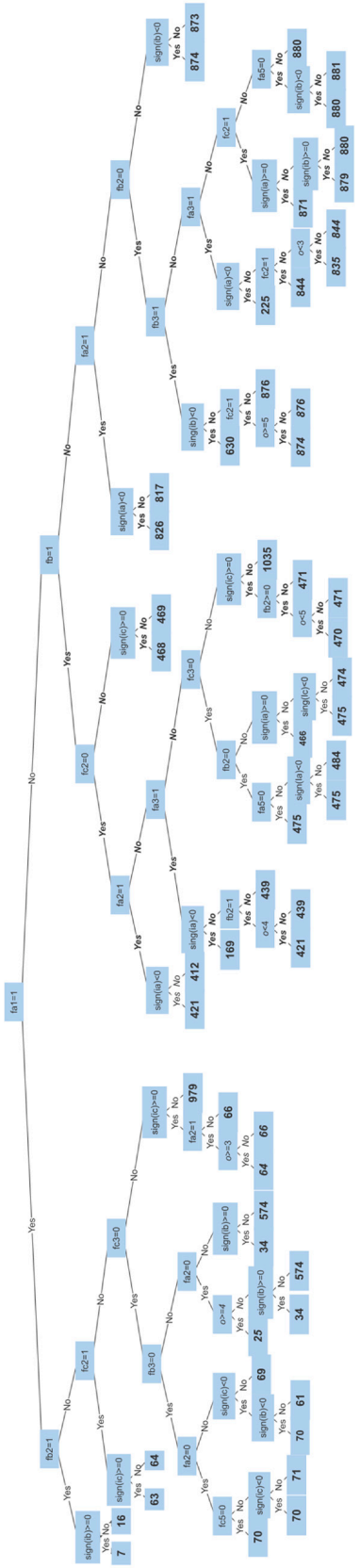


Fig. A.14. Example of a decision tree obtained for $v_{d1} > 0, v_{d2} > 0, v_{d3} > 0$ with the output shown in bold.

References

[1] Franquelo LG, León JI, Domínguez E. New trends and topologies for high power industrial applications: The multilevel converters solution. In: Proc. IEEE powereng. 2009, p. 1–6.
[2] Alves LFS, Gomes RCM, Lefranc P, De A. Pegado R, Jeannin P-O, Luciano B, et al. SIC power devices in power electronics: An overview. In: 2017 brazilian power electronics conference. 2017, p. 1–8.
[3] Lai J-S, Peng FZ. Multilevel converters-a new breed of power converters. IEEE Trans Ind App 1996;32(3):509–17.
[4] Tolbert LM, Shi X. Multilevel power converters. In: Rashid MH, editor. Power electronics handbook. 4th ed. Butterworth-Heinemann; 2018, p. 385–416, [ch. 13].
[5] Rodríguez J, Bernet S, Wu B, Pontt JO, Kouro S. Multilevel voltage-source-converter topologies for industrial medium-voltage drives. IEEE Trans Ind Electron 2007;54(6):2930–45.
[6] Rodríguez J, Bernet S, Steimer PK, Lizama IE. A survey on neutral-point-clamped inverters. IEEE Trans Ind Electron 2010;57(7):2219–30.
[7] Hotait H, Massoud A, Finney S, Williams B. Capacitor voltage balancing using redundant states of space vector modulation for five-level diode clamped inverters. IET Power Electron 2010;3(2):292.
[8] Zhao Z, Zhao J, Huang C. An improved capacitor voltage-balancing method for five-level diode-clamped converters with high modulation index and high power factor. IEEE Trans Power Electron 2016;31(4):3189–202.
[9] Cui D, Ge Q, Zhou Z, Yang B. A closed-loop voltage balance method for five-level diode-clamped inverters. In: 43rd annual conference of the IEEE industrial electronics society. 2017, p. 503–8.
[10] Montero-Robina P, Umbría F, Salas F, Gordillo F. Integrated control of five-level diode-clamped rectifiers. IEEE Trans Ind Electron 2019;66(9):6628–36.
[11] Vargas R, Cortes P, Ammann U, Rodriguez J, Pontt J. Predictive control of a three-phase neutral-point-clamped inverter. IEEE Trans Ind Electron 2007;54(5):2697–705.
[12] Gordillo F, Gómez-Estern F, Salas F. An optimization approach for modulation in multilevel converters. In: 42nd annual conference of the IEEE industrial electronics society. 2016, p. 5033–8.
[13] Montero-Robina P, Gordillo F, Gómez-Estern F, Salas F. Voltage balance for five-level DCC based on mixed-integer linear programming. Int J Electr Power Energy Syst 2021;124:106302.
[14] Narendra KS, Mukhopadhyay S. Neural networks in control systems. In: [1992] proceedings of the 31st IEEE conference on decision and control, vol. 1. 1992, p. 1–6.
[15] Lopez-Garcia TB, Coronado-Mendoza A, Domínguez-Navarro JA. Artificial neural networks in microgrids: A review. Eng Appl Artif Intell 2020;95:103894.
[16] Maruta H, Motomura M, Kurokawa F. A novel neural network based control method with adaptive on-line training for DC-DC converters. In: 2012 11th international conference on machine learning and applications, vol. 2. 2012, p. 503–8.
[17] Hajihosseini M, Andalibi M, Gheisarnejad M, Farsizadeh H, Khooban M-H. DC/DC power converter control-based deep machine learning techniques: real-time implementation. IEEE Trans Power Electron 2020;35(10):9971–7.
[18] Zhao S, Blaabjerg F, Wang H. An overview of artificial intelligence applications for power electronics. IEEE Trans Power Electron 2021;36(4):4633–58.
[19] Tang R, Zhang X. CART decision tree combined with boruta feature selection for medical data classification. In: 2020 5th IEEE international conference on big data analytics. 2020, p. 80–4.
[20] Breiman L. Random forests. Mach Learn 2001;5:5–32.
[21] Wang T, Pal A, Thorp JS, Wang Z, Liu J, Yang Y. Multi-polytope-based adaptive robust damping control in power systems using CART. IEEE Trans Power Syst 2015;30(4):2063–72.
[22] Gordillo F. A new modulation method for multilevel converters. In: 18th european conference on power electronics and applications. 2016.
[23] Liu J, Yin Y, Luo W, Vazquez S, Franquelo LG, Wu L. Sliding mode control of a three-phase AC/DC voltage source converter under unknown load conditions: industry applications. IEEE Trans Syst Man Cybern A 2018;48(10):1771–80.
[24] Yin Y, Liu J, Luo W, Wu L, Vazquez S, Leon JI, et al. Adaptive control for three-phase power converters with disturbance rejection performance. IEEE Trans Syst Man Cybern A 2018;1–12.
[25] Vázquez S, Sánchez Ja, Carrasco JM, León JI, Galván E. A model-based direct power control for three-phase power converters. IEEE Trans Ind Electron 2008;55(4):1647–57.
[26] Montero-Robina P, Rouzbehi K, Gordillo F, Pou J. Grid-following voltage source converters: basic schemes and current control techniques to operate with unbalanced voltage conditions. IEEE Open J Ind Electron Soc 2021;1.
[27] Yazdani A, Reza I. Controlled dc-voltage power port. In: Voltage-sourced converters in power systems: modeling, control, and applications. Hoboken: IEEE Press and John Wiley & Sons; 2010, p. 189–203, [ch. 7.5].
[28] Teodorescu R, Blaabjerg F, Liserre M, Loh PC. Proportional-resonant controllers and filters for grid-connected voltage-source converters. IEE Proc-Electr Power Appl 2006;153(5):750–62.