electronics

MDPI

Article

NN-Based Approach to Model Predictive Control for Three-Phase Inverters Using One- or Two-Step Horizons

Abdul Aziz Zazai and Yuan Gao *

School of Engineering, University of Leicester, Leicester LE1 7RH, UK; aaz8@student.le.ac.uk
* Correspondence: yuan.gao@leicester.ac.uk

Abstract

Finite-control-set model predictive control (FCS-MPC) gives low-distortion voltage regulation for LC-filtered two-level three-phase inverters, but its computation grows exponentially with the prediction horizon. We trained neural networks offline to imitate it. Four imitators were formed by crossing two network types, a pattern-recognition classifier (NNPR) and a regression network (NNRG), with data from one-step and two-step FCS-MPC teacher controllers, giving NNPR1, NNPR2, NNRG1, and NNRG2. All were evaluated in MATLAB/Simulink with linear resistive loads at sixteen training and four unseen interpolated conditions. Mean output-voltage total harmonic distortion (THD) was 2.55% for NNPR1, 0.89% for NNPR2, 2.89% for NNRG1, and 1.36% for NNRG2, against 2.32% and 1.73% for one-step and two-step FCS-MPC, respectively. The training horizon mattered more than the network type, and the classifier gave lower mean THD at both horizons. NNPR2 gave lower THD than its teacher controller at all sixteen conditions while switching 26% less often. The advantage came mainly from the load current, which the teacher controller estimated but the networks received as measured; given the measured value, the teacher controller came within 0.13 percentage points of NNPR2. Network computation is horizon-independent; MATLAB execution times are indicative only. Results are simulation-only; hardware-in-the-loop validation with delay compensation must precede deployment.

Keywords: finite-control-set model predictive control; neural network control; two-level voltage source inverter; LC filter; total harmonic distortion; regression; pattern recognition; measurements

1. Introduction

Two-level three-phase voltage-source inverters (VSIs) are widely used to convert a DC supply into a controllable AC output in applications including uninterruptible power supplies (UPSs) [1], distributed generation [2], grid-connected photovoltaic systems [3], and conversion systems in which a three-phase supply must be produced from a single-phase source [4]. Among the available inverter topologies, the two-level bridge remains the dominant choice at low voltage owing to its structural simplicity and low component count [5], whereas multilevel topologies are generally reserved for medium- and high-power applications in which their reduced harmonic content and lower device stress justify the additional circuit and modulation complexity [5,6]. In these applications, a high-quality sinusoidal output voltage with low total harmonic distortion (THD) is required, particularly for nonlinear or unbalanced loads [1,3]. An output LC filter is commonly inserted between the inverter bridge and the load to attenuate switching-frequency harmonics and to limit

Check for updates

Academic Editors: Samir Abood, Annamalai Annamalai and Mohamed F. Chouikha

Received: 13 August 2026
Revised: 31 October 2026
Accepted: 2 November 2026
Published: 5 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

the rate of change of the output voltage [7], producing a low-distortion waveform at the cost of a more demanding controller design [1,7–9].

Many linear and nonlinear control schemes have been proposed for the LC-filtered VSI, including hysteresis control, proportional–integral regulators combined with pulse-width modulation (PWM) or space-vector modulation (SVM), deadbeat control, multi-loop feedback control, and sliding-mode control [1,8,10]. In the majority of these schemes, nested voltage and current loops operate together with an external modulator that generates the switching pulses [1,10], and the choice of modulation strategy itself influences the harmonic quality of the output, with space-vector PWM offering better DC-bus utilisation and lower THD than sinusoidal PWM [11].

Finite-control-set model predictive control (FCS-MPC) has emerged as an attractive alternative owing to its fast dynamic response and its capacity to incorporate system constraints and nonlinearities directly within the controller design [1,12,13]. At each sampling instant, FCS-MPC uses a discrete-time model of the system to predict the output for every admissible switching state and applies the state that minimises a cost function which penalises the predicted voltage-tracking error, thus removing the need for a modulator and for cascaded control loops, leading to high-quality power control [1,12]. This exhaustive evaluation imposes a substantial online computational load, and several methods have been proposed to reduce or compensate for it, including simplified formulations that reduce the number of candidate states evaluated per sampling period [14] and delay-compensation schemes that account for the computation time within the prediction itself [15,16].

Extending the prediction horizon beyond a single step improves control quality. Cortés et al. showed that increasing the horizon of an LC-filtered three-phase UPS inverter from one step to two steps reduced the output-voltage THD under a resistive load from 2.15% to 1.56% when a different vector was evaluated in each step, at the cost of additional computation [17]. With seven distinct candidate vectors per step, a two-step horizon requires the evaluation of $7 \times 7 = 49$ switching sequences, and the number of sequences grows exponentially with horizon length [17–19]. Cortés et al. also applied the same vector over both steps, which limits the search to seven sequences; this variant gave a similar THD of 1.54% [17]. More generally, suboptimal search strategies with bounded computational complexity [18] and alternative optimisation algorithms for the multi-step FCS-MPC of two-level inverters [20,21] have been investigated to contain the cost of longer horizons; these methods reduce, but do not remove, an online optimisation burden that grows with the horizon, and multi-step FCS-MPC therefore remains a barrier to real-time embedded implementation [17–21].

To decouple control quality from this online cost, data-driven approaches have recently been investigated in which an artificial neural network (ANN) is trained offline to imitate an FCS-MPC controller and then replaces it at run time; because the execution time of the network depends only on its size, and not on the horizon or the number of switching states of the imitated controller, the deployment cost is independent of the complexity of the policy being imitated [22–25]. Mohamed et al. applied this paradigm to a two-level three-phase VSI with an output LC filter, training a feed-forward classification (pattern-recognition) network on one-step FCS-MPC data; the network receives the $\alpha\beta$ components of the filter current, capacitor voltage, output current, and reference voltage as eight inputs and produces a one-hot output identifying the optimal voltage vector among the seven candidates [23]. The reported network attained a THD comparable to or lower than that of its one-step FCS-MPC teacher controller across several operating conditions while reducing the online computation, and the present work builds directly on the two-level VSI MATLAB/Simulink models introduced in that study [23]. Novak and Dragičević demonstrated a related imitation learning scheme, validating ANN imitators of FCS-MPC on embedded hardware and noting

that the imitator is intended to reproduce, rather than surpass, the performance of the original controller at a comparable switching frequency [22]. Two neural network types have been used for such imitators: a regression formulation—in which the switching decision is produced as a continuous quantity that is subsequently quantised to an admissible state—and a classification formulation, in which the discrete switching state is selected directly through a softmax output [22,26,27]. Wang et al. compared both formulations as imitators of FCS-MPC for a modular multilevel converter (MMC) and reported that the regression network achieved better control performance than the pattern-recognition network; however, the accuracy of the pattern-recognition network was limited by the under-representation of certain switching classes in the MMC training data rather than by the neural network type itself [26]. Learning can also be combined with MPC without replacing the optimiser. Cavus et al. fed machine-learning forecasts into a chance-constrained stochastic MPC of a multi-energy microgrid, so the optimiser kept its constraint handling while the learned component improved its predictions [28]. The present work takes the opposite approach: the network replaces the online optimisation and the explicit constraint handling of FCS-MPC is given up at run time (Section 4.5).

FCS-MPC is intrinsically a discrete selection problem, which the classification formulation represents without quantisation. In the two-level VSI, the eight switching combinations reduce to seven classes—because the two zero vectors are merged—so the six active vectors are near-uniformly represented in the training data and the severe class-imbalance limitation reported for the MMC does not arise from the topology itself [23,26]; the merged null state remains the one sparsely populated class, and its share is set by the prediction horizon of the teacher controller rather than by the converter. Two questions therefore remain open for the two-level VSI with an output LC filter: whether the classification or the regression formulation is preferable for this topology and whether the established control-quality advantage of two-step over one-step FCS-MPC [17] is transferred to a network that imitates it. To address these questions, this paper reports a 2 × 2 ablation that crosses the network formulation—classification, following Mohamed et al. [23], and regression, following Wang et al. [26]—with the prediction horizon of the FCS-MPC controller used to generate the training data, one step, as in [23], and two steps, as in [17]. This produces four imitation controllers: (NNPR1) a classification network trained on one-step data, which reproduces the configuration of Mohamed et al. [23] and serves as the baseline, although the exact same model is not used; (NNPR2) a classification network trained on two-step data; (NNRG1) a regression network trained on one-step data; and (NNRG2) a regression network trained on two-step data.

The central hypothesis is that embedding a richer underlying control policy in the training data improves the control quality of the imitator. Because two-step FCS-MPC is a superior policy to one-step FCS-MPC in terms of THD [17], a network trained on two-step data is expected to inherit that relative advantage, such that NNPR2 improves upon NNPR1 and NNRG2 improves upon NNRG1. The ablation is constructed so that each comparison isolates a single factor: comparing NNPR1 with NNRG1 and NNPR2 with NNRG2 isolates the effect of network architecture (classification versus regression) at a fixed teacher policy, whereas comparing NNPR1 with NNPR2 and NNRG1 with NNRG2 isolates the effect of embedded policy richness (one-step versus two-step horizon) at a fixed architecture. The imitation of a two-step FCS-MPC policy by either network formulation has not previously been reported for a two-level VSI with an output LC filter, the prior work on this topology being restricted to one-step classification [23] and the regression-versus-classification comparison having been conducted only for the MMC [26].

The four controllers are evaluated against their FCS-MPC teacher controllers across sixteen operating conditions, defined by sweeps of load resistance, DC-link voltage, and

reference voltage, and at four interpolated unseen conditions, using the following control-quality metrics: the steady-state output-voltage THD, defined as the ratio of the root-mean-square of the harmonic components to the root-mean-square of the fundamental; the transient voltage-tracking error, quantified as the root-mean-square error (RMSE) between the reference and the measured output voltage; the settling time following a step change in the operating point; and the per-sample execution time, which characterises the online computational cost [22,23]. Section 2 presents the system model together with the one- and two-step FCS-MPC formulations; Section 3 describes the neural network controllers, including data collection, preparation, architecture, training, and deployment; Section 4 reports the comparative results across the four controllers and the FCS-MPC baselines; Section 5 discusses the mechanisms behind those results and the limitations that bound them; and Section 6 summarises the findings and identifies future study.

The main contributions of this paper are summarised as follows:

- The imitation of a two-step FCS-MPC policy by a neural network is reported for the first time for a two-level VSI with an output LC filter, in both classification and regression form, combining the one-step classification imitator presented in [23] with the two-step teacher policy presented in [17].
- A controlled 2 × 2 ablation is constructed that separates the effect of the neural network type (classification versus regression) from that of the training-data FCS-MPC horizon (one step versus two steps).
- The generalisation of the trained controllers is quantified at unseen operating conditions interpolated within the training envelope, and the per-sample execution times of the network and FCS-MPC controllers are compared, characterising the accuracy–cost trade-off relevant to embedded deployment [17,22].

2. System Fundamentals

This section presents the mathematical model of the power electronic circuit shown in Figure 1. The model of the LC filter is also described. These models are used for the FCS-MPC of the inverter.

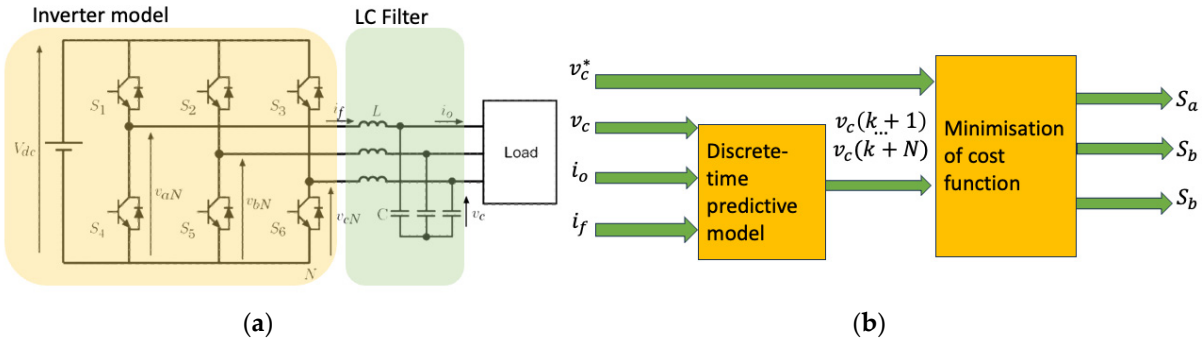


Figure 1. (**a**) Topology of the three-phase inverter model with output LC filter and (**b**) general MPC process.

2.1. Inverter Topology and Switching States

The DC bus is supplied by a constant voltage V_{dc}. The three-phase inverter (Figure 1) is made of three legs, which each contain an upper and lower power switch pair [5]. Each leg is connected to a phase of the output LC filter, which then supplies the load. In this paper, only linear loads are used for generating training data.

The two switches situated within each leg are operated in a complementary manner, i.e., when the upper switch is ON, the lower switch is OFF, and vice versa. If both switches in a single leg are ON, a short circuit results. Hence, the switching state of the inverter can

be represented by a binary switching signal. Let the switching state be S_n where $n \in \{a, b, c\}$; therefore,

$$S_a = \begin{cases} 1, & \text{if } S_1 \text{ ON and } S_4 \text{ OFF} \\ 0, & \text{if } S_1 \text{ OFF and } S_4 \text{ ON} \end{cases}$$

$$S_b = \begin{cases} 1, & \text{if } S_2 \text{ ON and } S_5 \text{ OFF} \\ 0, & \text{if } S_2 \text{ OFF and } S_5 \text{ ON} \end{cases}$$

$$S_c = \begin{cases} 1, & \text{if } S_3 \text{ ON and } S_6 \text{ OFF} \\ 0, & \text{if } S_3 \text{ OFF and } S_6 \text{ ON} \end{cases}$$

To reduce the three-phase system into two-dimensional representations, circuit quantities are put into the αβ reference frame using the Clarke transformation. Therefore, the three binary switching states are combined into a switching vector:

$$S = \frac{2}{3}\left(S_a + aS_b + a^2S_c\right) \equiv S_\alpha + jS_\beta \tag{1}$$

where $a = e^{j\frac{2\pi}{3}}$ is the complex operator which corresponds to the 120° phase difference between the legs. Using (1), in matrix form the transformation is written as

$$\begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \tag{2}$$

Because each of the three legs accepts two states, there are $2^3 = 8$ distinct switching combinations. Using (2) for all eight switching combinations results in two redundant zero vectors: v_0, which corresponds to (0,0,0) where all the lower switches are ON, and v_1, which corresponds to (1,1,1) where all the upper switches are ON. The vectors v_0 and v_1 produce zero voltage and are therefore combined into a single vector v_7. The remaining six active vectors are distributed symmetrically at 60° intervals in the αβ reference frame (Figure 2). The output-voltage space vectors produced by the inverter can be defined by

$$v_i = \frac{2}{3}\left(v_{aN} + av_{bN} + a^2v_{cN}\right) \tag{3}$$

where v_{aN}, v_{bN}, and v_{cN} are the phase voltages of the inverter with reference to N (see Figure 1), which is the negative terminal of the DC link. The voltage vector v_i in terms of V_{dc} and switching-state vector S can be defined by

$$v_i = V_{dc}S \tag{4}$$

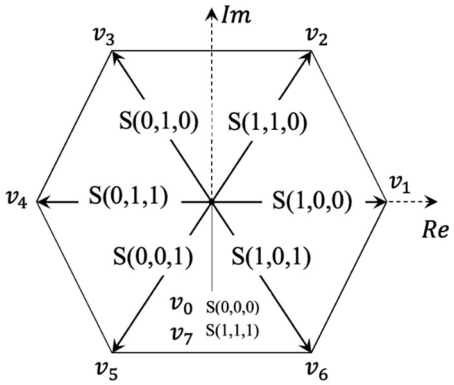


Figure 2. All possible switching vectors that can be applied to the two-level, three-phase inverter and their related switching states in the αβ reference frame.

Using vectorial form, the filter current i_f, the output voltage v_c, and the output current i_o can be expressed as space vectors in the αβ reference frame in the following way:

$$i_f = \frac{2}{3}\left(i_{fa} + a i_{fb} + a^2 i_{fc}\right) \equiv i_{f\alpha} + j i_{f\beta} \tag{5}$$

$$v_c = \frac{2}{3}\left(v_{ca} + a v_{cb} + a^2 v_{cc}\right) \equiv v_{c\alpha} + j v_{c\beta} \tag{6}$$

$$i_o = \frac{2}{3}\left(i_{oa} + a i_{ob} + a^2 i_{oc}\right) \equiv i_{o\alpha} + j i_{o\beta} \tag{7}$$

2.2. LC Filter Dynamics

There are two equations which can be used to model the LC filter:

$$L\frac{di_f}{dt} = v_i - v_c \tag{8}$$

$$C\frac{dv_c}{dt} = i_f - i_o \tag{9}$$

where L and C are the filter inductance and filter capacitance respectively.

The state-space representation of these equations is as follows:

$$\frac{dx}{dt} = Ax + Bv_i + B_d i_o \tag{10}$$

where

$$x = \begin{bmatrix} i_f \\ v_c \end{bmatrix} \tag{11}$$

$$A = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \tag{12}$$

$$B = \begin{bmatrix} -\frac{1}{L} \\ 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ -\frac{1}{C} \end{bmatrix} \tag{13}$$

The outputs from the inverter i_f and V_c are measured; i_o is a disturbance that can be measured or approximately determined using an observer [1]. The value of V_{dc} is assumed to be fixed and known in this paper. The FCS-MPC controllers in this work do not measure i_o; they estimate it from the sampled i_f and v_c (Section 2.4.2).

The output voltage v_c can be represented as a state equation:

$$v_c = \begin{bmatrix} 0 & 1 \end{bmatrix} x \tag{14}$$

Voltage vector v_i is the input to the system of the inverter in Figure 1, which is the system being represented by (14), and is part of a finite set of seven voltage vectors which correspond to a switching state, as shown in Figure 2.

2.3. Discrete-Time Model of the LC Filter

Using (10), a discrete-time model of the output LC filter can be obtained for sampling period T_s:

$$x(k+1) = A_q x(k) + B_q v_i(k) + B_{dq} i_o(k) \tag{15}$$

where

$$A_q = e^{AT_s} \tag{16}$$

$$B_q = \int_0^{T_s} e^{A\tau} B \, d\tau \tag{17}$$

$$B_{dq} = \int_0^{T_s} e^{A\tau} B_d \, d\tau \tag{18}$$

Equation (14) is used to determine predictions of output voltage V_c for the given voltage vector v_i. This discrete-time model is used as a prediction engine within FCS-MPC, which is presented in the next section.

2.4. Model Predictive Control

Using (15), predictions of the output voltage are calculated until time k + N, where N is the number of prediction steps, for a given number of measurements and sequence of voltage vectors at time k. The scheme uses a cost function that minimises the error between the predicted voltage and the reference voltage v_c^*. The overall scheme is shown in the lower part of Figure 1.

The cost function J_n, where n is the number of steps of prediction, is the sum of the squared errors between the predicted and reference-voltage vectors from time k + 1 to k + N. Thus, the cost function for one-step FCS-MPC with one-step prediction can be defined as follows:

$$J_1 = \left(v_{c\alpha}^* - v_{c\alpha}^p(k+1)\right)^2 + \left(v_{c\beta}^* - v_{c\beta}^p(k+1)\right)^2 \tag{19}$$

where $v_{c\alpha}^*$ and $v_{c\beta}^*$ are the reference-voltage vector's real and imaginary parts, and $v_{c\alpha}^p(k+1)$ and $v_{c\beta}^p(k+1)$ are the real and imaginary part of the predicted voltage vectors, which were calculated using (14).

2.4.1. Extending the Horizon of Model Predictive Control

For two-step prediction, the cost function is

$$J_2 = \left(v_{c\alpha}^* - v_{c\alpha}^p(k+1)\right)^2 + \left(v_{c\beta}^* - v_{c\beta}^p(k+1)\right)^2 + \left(v_{c\alpha}^* - v_{c\alpha}^p(k+2)\right)^2 + \left(v_{c\beta}^* - v_{c\beta}^p(k+2)\right)^2 \tag{20}$$

In one-step FCS-MPC, only seven voltage vectors are applied during one sampling period. On the other hand, during two-step FCS-MPC, a voltage vector is applied during the first sampling period and then another is applied during the second. This leads to $7^2 = 49$ possible sequences that need to be evaluated by the cost function. This is shown in Figure 3. To reduce the computation, Cortés et al. also applied the same voltage vector in both sampling periods, which cuts the number of sequences from 49 to 7 [17]. We do not use this simplification: the two-step controller in this paper evaluates all 49 sequences, which corresponds to the different-vector case in [17] (1.56% THD under a resistive load, against 1.54% with the same vector). The full search gives the networks the unrestricted two-step decision as their training label, at a higher online cost for the FCS-MPC baseline.

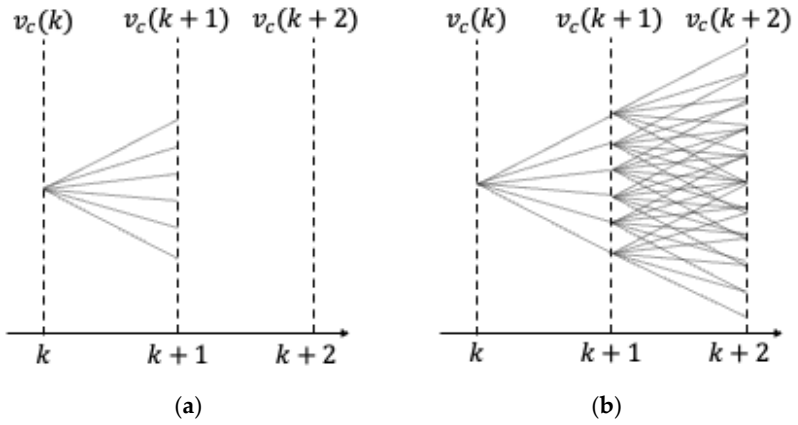


Figure 3. Output voltage prediction trajectories for different prediction horizons. (**a**) One-step prediction. (**b**) Two-step prediction.

As previously established in [17], two-step FCS-MPC produces superior output-voltage quality compared with one-step FCS-MPC. However, due to the number of calculations required, the computational burden of two-step FCS-MPC is higher than one-step FCS-MPC. To address this, two-step FCS-MPC is used to train a neural network offline; the neural

network is trained to replicate the FCS-MPC's input-output mapping. This results in the high-quality performance of two-step FCS-MPC while reducing its computational burden. The proposed neural network controllers are presented in the following section.

2.4.2. Load-Current Estimation

The predictions (15) require the load current $i_o(k)$, which acts on the LC filter as a disturbance (Section 2.2). The FCS-MPC controllers in this work do not measure it; instead, they estimate it at every sampling instant from the filter current and the capacitor voltage by applying a backward difference to (9):

$$i_o(k) = i_f(k-1) - \frac{C}{T_s}[v_c(k) - v_c(k-1)] \tag{21}$$

where $i_o(k)$ is the estimated load-current space vector at sampling instant k; $i_f(k-1)$ is the filter-current space vector measured at the previous sampling instant; $v_c(k)$ and $v_c(k-1)$ are the capacitor-voltage space vectors measured at the present and previous sampling instants; C is the filter capacitance; and T_s is the sampling period (Table 1). The term $(\frac{C}{T_s})[v_c(k) - v_c(k-1)]$ approximates the capacitor current $C\frac{dv_c}{dt}$ of (9) over the last sampling period, and subtracting it from the filter current leaves the current that flows into the load. The estimate is computed once per sampling period and is held constant over the prediction horizon, so the same value of i_o is used in both steps of a two-step prediction. Because the voltage difference is multiplied by $\frac{C}{T_s} = 40\ \mu F/30\ \mu s \approx 1.33$ A/V, every 1 V of sample-to-sample change in v_c, including the switching ripple, changes the estimate by about 1.33 A, which is comparable with the 2 A peak load current at the baseline condition of Table 1. The estimate therefore carries part of the voltage ripple. Section 4.7 examines how this affects the teacher controllers and the networks trained on their data.

Table 1. Baseline parameters for linear loads.

Parameter	Value
Sampling time, T_s	30 [μs]
Filter inductance, L	2.5 [mH]
Filter capacitance, C	40 [μF]
DC-link voltage, V_{dc}	500 [V]
Load resistance, R	100 [Ω]

3. Neural Network Controllers

The deterministic nature of FCS-MPC means that for any given number of system measurements there will be a given output. The mapping from the measured state to the optimal switching vector is therefore a fixed, memoryless function that a feed-forward neural network can approximate offline and then evaluate online at a fixed cost, without solving the optimisation at each sampling instant [22,26]. The process is shown in Figure 4. Two neural network formulations are examined. The first, a neural network pattern-recognition controller (NNPR), treats the selection of the switching vector as a classification problem and is trained to reproduce the discrete decision of FCS-MPC directly. The second, a neural network regression controller (NNRG), treats the same selection as a regression onto a continuous output from which a discrete decision is determined. Because FCS-MPC selects from a finite set of switching states, its decision is inherently categorical; comparing the two formulations helps us determine whether preserving that categorical structure, rather than regressing through a continuous output, affects control quality. Each formulation is trained separately on data generated by one-step and by two-step FCS-MPC, yielding the

four neural controllers evaluated in Section 4 and the 2 × 2 comparison of network type against training horizon.

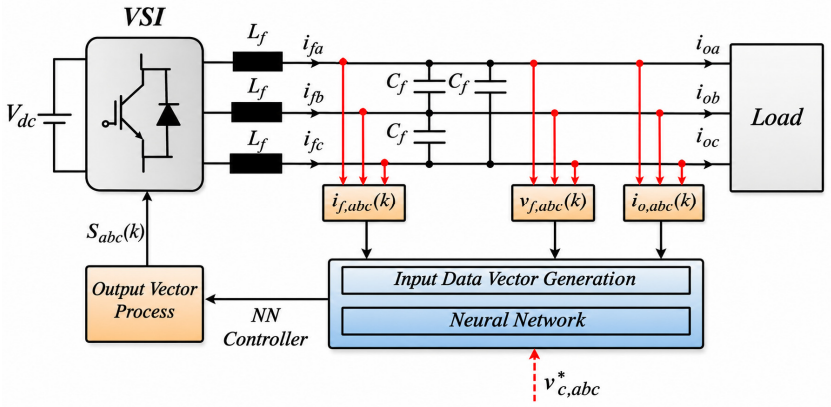


Figure 4. Procedure for generating training data from FCS-MPC and for training and deploying the neural network controllers.

3.1. Data Collection

The training data are generated by running the FCS-MPC controllers in MATLAB/Simulink across the sixteen operating conditions listed in Table 2, in which the load resistance R, the DC-link voltage V_{dc}, and the reference-voltage amplitude v_c^* are swept over the ranges used throughout this work. Sweeping these conditions ensures that the dataset spans the operating envelope over which the network is later required to act, so that the learned mapping is exercised across the full range of states the controller will encounter rather than at a single operating point [22]. The one-step and two-step controllers are run separately, producing two parallel datasets that differ only in the prediction horizon of the teacher controller that labelled them. After the cleaning described in Section 3.2, each of the sixteen runs spans 0.5 s of simulated operation—that is, twenty-five fundamental cycles at 50 Hz—and contributes 16,567 labelled samples once the start-up transient is discarded. Each teacher controller therefore yielded 265,072 labelled samples, the two totals coinciding because both controllers were run over the same sixteen operating conditions at the same duration and sampling period; the datasets differ in the trajectories visited and in the switching-state labels assigned, not in their size. Each sample comprises the eight-dimensional input vector and the corresponding optimal switching-state index. All sixteen conditions use a balanced three-phase linear resistive load; the training data contain no nonlinear or unbalanced load.

At each sampling instant, the state of the system is recorded. The capacitor voltage v_c^*, the filter current i_f, the load current i_o, and the reference voltage v_c^* are logged as complex quantities in αβ coordinates, and the optimal switching-state index selected by the cost function at that instant is logged alongside them as the target. Each complex αβ signal is separated into its real and imaginary parts, giving an eight-dimensional input feature vector $[v_c, v_{c\beta}, i_f, i_{f\beta}, v_c^*, v_{c\beta}^*, i_o, i_{o\beta},]$; the corresponding target is the single switching-state index in {1, ..., 7} that FCS-MPC found to be optimal.

The load-current input i_o in these data is the estimate (21) used by the teacher controllers, so every sample contains it, both as an input and through its label. In closed-loop operation (Section 4), however, the networks receive the measured load current; Section 4.7 examines the consequences.

Table 2. A comparison of THD of output voltage between one-step FCS-MPC, two-step FCS-MPC, and the proposed NNPR and NNRG controllers under training operating conditions. THD is computed for phase A of the capacitor voltage v_c. Fixed parameters: $Ts = 30$ µs, $f = 50$ Hz, $C = 40$ µF, $L = 2.5$ mH. Bold results show which controller outperformed its classical counterpart.

Sample No.	R [Ω]	Vdc [V]	vc^* [V]	THD [%]					
				One-Step FCS-MPC	Two-Step FCS-MPC	NNPR (One-Step)	NNPR (Two-Step)	NNRG (One-Step)	NNRG (Two-Step)
Baseline operating conditions (Vdc = 500 V, vc* = 200 V)									
S1	5	500	200	1.24	1.41	**0.96**	**0.69**	1.14	**1.36**
S2	10	500	200	1.63	1.57	1.66	**0.79**	1.55	**1.20**
S3	20	500	200	1.90	1.61	2.43	**0.76**	2.86	**1.30**
S4	30	500	200	2.32	1.69	2.91	**0.98**	2.59	**1.39**
S5	50	500	200	2.90	1.95	4.13	**0.99**	3.01	**1.53**
S6	80	500	200	2.27	1.90	2.65	**0.90**	2.62	**1.29**
S7	100	500	200	3.11	1.15	3.16	**1.01**	3.69	**1.40**
S8	200	500	200	2.26	1.98	2.64	**0.97**	2.90	**1.22**
S9	500	500	200	2.62	2.07	3.24	**0.96**	4.67	**1.37**
S10	1000	500	200	3.18	1.99	2.71	**0.87**	3.71	**1.38**
Reduced DC-link and reference (Vdc = 400 V, vc* = 150 V)									
S11	10	400	150	1.58	1.58	1.62	**0.79**	1.81	**1.22**
S12	50	400	150	2.23	1.72	2.25	**0.92**	3.76	**1.25**
S13	100	400	150	2.73	2.11	2.83	**1.01**	3.15	**1.29**
Elevated DC-link and reference (Vdc = 600 V, vc* = 250 V)									
S14	10	600	250	1.71	1.59	**1.49**	**0.69**	2.11	**1.31**
S15	50	600	250	2.54	1.55	2.87	**0.94**	3.17	**1.64**
S16	100	600	250	2.82	1.88	3.22	**1.01**	3.44	**1.58**

3.2. Data Preparation

Before training, the merged dataset is cleaned in three stages. Firstly, the initial one hundred samples of every run are discarded. These samples belong to the start-up transient state, during which the output voltage v_c has not yet reached the reference voltage v_c^* and the switching decisions are unrepresentative of steady-state operation; retaining them would bias the learned mapping toward a regime that does not recur once the VSI is operating normally. Secondly, any sample whose logged index falls outside the valid range of one to seven is removed, which guards against spurious labels introduced by logging artefacts at the run boundaries. Thirdly, the per-class sample counts are inspected to confirm that all seven switching states are represented. This check matters because the pattern-recognition formulation relies on a softmax classifier, which is sensitive to class imbalance: a state that appears rarely in the training data is learned poorly, as can be seen from the results of Wang et al. [26]. In the modular-multilevel application of Wang et al. [26], certain insert-number classes for the MMC were under-represented, which limited the NNPR's accuracy. The two-level, three-phase inverter studied here exposes only seven switching states, all of which are exercised across the load sweep. The six active vectors are near-uniformly represented, each accounting for between 16.24% and 16.56% of the one-step dataset, so the severe under-representation observed by [26] does not arise for the active vectors. The merged null state is the exception, accounting for 1.50% of the one-step dataset. Under the two-step teacher controller, the same six active vectors span 15.14% to 15.43% and the null state rises to 8.12%, which confirms that the population of this class is governed by

the prediction horizon of the teacher controller rather than by the converter topology; the consequences are examined in Section 3.4.

The eight input features are then normalised to the range $[0, 1]$ using a min–max mapping. Normalisation is necessary because the raw features differ by more than an order of magnitude in scale—the voltages are on the order of hundreds of volts, whereas the currents are on the order of tens of amperes—so without it the larger-magnitude features would dominate the gradient and be weighted disproportionately by the neural network. The minimum and maximum used for the mapping are retained and applied unchanged when the network is deployed, which guarantees that an input presented at run time is scaled exactly as the training data were.

The same cleaned and normalised dataset is used for both the NNPR and the NNRG networks; only the representation of the target differs between them, as described in Section 3.3. Holding the input data fixed in this way ensures that any difference in performance between the two formulations is attributable to the formulation itself rather than to a difference in the data each NN is trained on.

3.3. Network Architecture

Both controllers use a shallow feed-forward network (shown in Figure 5) with a single hidden layer: eight inputs, fifteen neurons in the hidden layer with a hyperbolic-tangent-sigmoid activation, and seven outputs. A single hidden layer is sufficient because such a network is a universal approximator of the bounded input-output mapping required here, and keeping the network shallow keeps the deployed forward pass small. The number of hidden neurons was set using the rule of thumb adopted by Wang et al. [26], which bounds the count between a minimum of $0.5(N_{in} + N_{out}) = 7.5$ and a maximum of $2N_{in} = 16$, where $N_{in} = 8$ and $N_{out} = 7$ are the input and output unit counts; fifteen neurons were chosen, near the upper end of this range, to provide sufficient capacity to fit the mapping while remaining within the recommended bound.

The two formulations differ only in the output layer and in the quantity each is trained to reproduce. In the NNPR network the output layer applies a softmax function, so that the seven outputs are constrained to lie in $(0, 1)$ and the sum results in 1; each output is then interpreted as the probability that its switching state is optimal, and the state with the highest probability is applied to the inverter. The softmax function couples the seven outputs through a competitive normalisation, which matches the structure of the FCS-MPC decision, where exactly one switching state is selected at each instant. In the NNRG network the output layer is linear and the targets are the one-hot encoding of the optimal index treated as continuous values; each output is fitted independently under a squared-error criterion, and the applied switching state is taken as the index of the largest output. Because the underlying decision is categorical, this regression introduces a quantisation step that the classification formulation avoids (the continuous outputs must be reduced to a discrete, admissible index) and the independent fitting of each output does not enforce the mutual exclusivity that the softmax imposes. Comparing the two networks therefore tests directly whether preserving the categorical structure of the FCS-MPC decision yields better control.

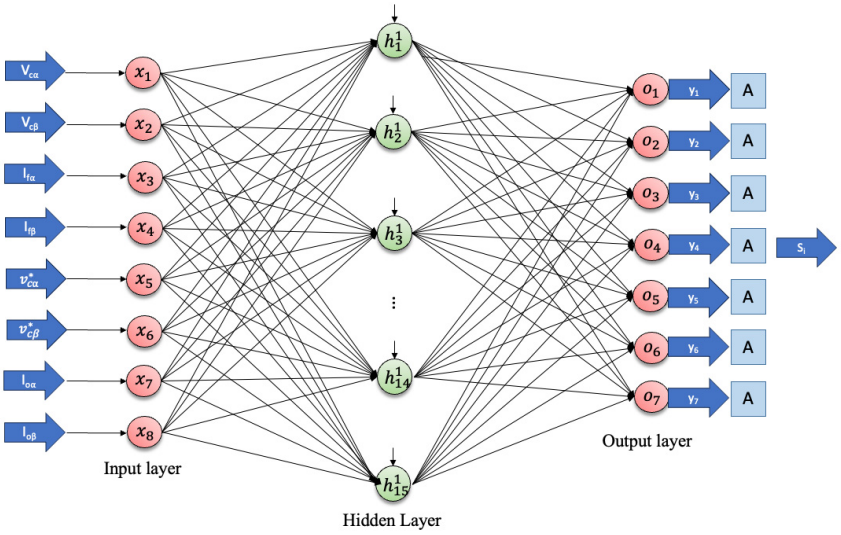
Each controller is expressed as a vector-valued function of the eight-dimensional input feature vector in $\alpha\beta$ coordinates. The pattern-recognition network (NNPR) is written as

$$(P(S_1), \ldots, \mathrm{P}(S_7)) = F\left(v_{c\alpha}, v_{c\beta}, i_{f\alpha}, i_{f\beta}, v^*_{c\alpha}, v^*_{c\beta}, i_{o\alpha}, i_{o\beta}\right) \tag{22}$$

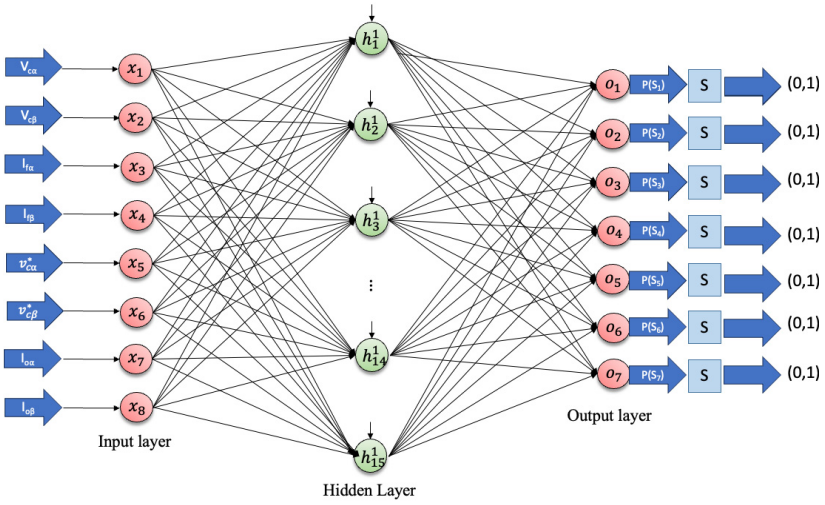
where F denotes the classifier and the outputs as the probabilities that S_i, where $i \in \{1, \ldots, 7\}$, is the optimal switching state, which sum to unity; the inverter applies the state with the highest probability. The regression network (NNRG) takes the same inputs and is written as

$$(y_1, \ldots, y_7) = f\left(v_{c\alpha}, v_{c\beta}, i_{f\alpha}, i_{f\beta}, v^*_{c\alpha}, v^*_{c\beta}, i_{o\alpha}, i_{o\beta}\right) \tag{23}$$

where f denotes the regressor, the continuous outputs y_1, ..., y_7 are fitted to the one-hot encoding of the optimal index under a squared-error criterion, and the applied state S_i, where $i \in \{1, \ldots, 7\}$, is again taken as the index of the largest output. The two networks therefore represent the same deterministic FCS-MPC mapping [22,26] and differ only in whether it is realised as a classification, F, or a regression, f.



(**a**)



(**b**)

Figure 5. (**a**) NNRG architecture; A stands for argmax, which selects the largest output value. (**b**) NNPR architecture; S stands for softmax, which outputs the highest probability of the switching state S_i with $i \in \{1, \ldots, 7\}$ as 1 and the remaining as 0 s, selecting only the highest probability of the given switching state as being optimal.

3.4. Training

The two datasets of one- and two-step FCS-MPC are each partitioned into a training set (70%), a validation set (15%), and a test set (15%) by a random permutation of the samples under a fixed random seed. Fixing the seed makes the partition reproducible and, because the same saved partition is loaded for both networks, guarantees that the NNPR and NNRG controllers are trained, validated, and tested on identical samples, so the comparison between them is controlled. The validation set is used to terminate training, and the test set—which is not seen during training—provides an estimate of generalisation within the training data envelope.

Both networks are trained with the scaled conjugate-gradient algorithm. This method is chosen because it retains the fast convergence of conjugate-gradient optimisation while requiring neither a line search nor any user-selected learning-rate parameters, which makes the training procedure robust and repeatable [22]. Training proceeds for up to one thousand epochs but is stopped early if the validation error fails to improve for fifty consecutive epochs; early stopping prevents over-fitting, since training stops once performance on the held-out validation data ceases to improve. The NNPR network is trained under a cross-entropy loss appropriate to its softmax classifier, whereas the NNRG network is trained under a mean-squared-error loss appropriate to its continuous output.

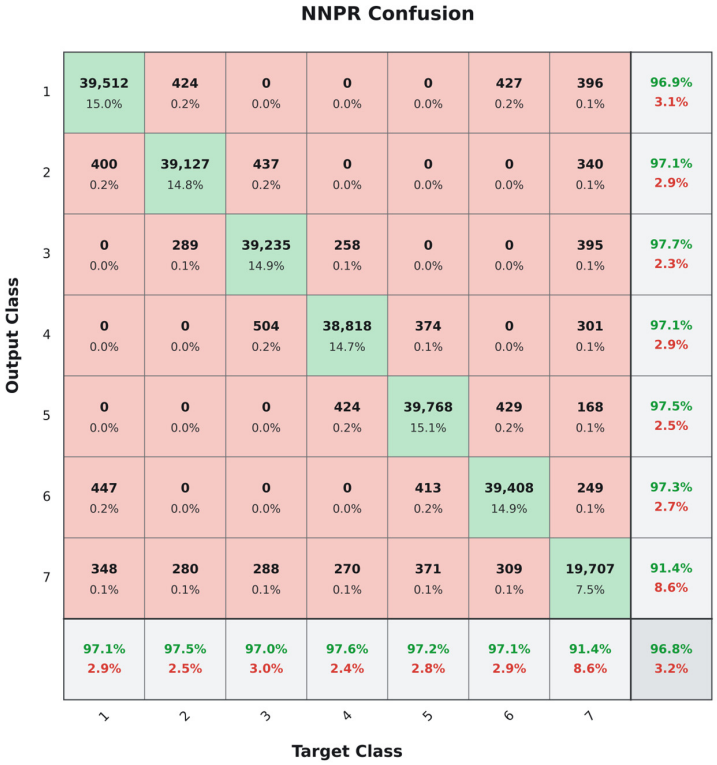
Figure 6 reports the confusion matrices of the four imitation networks, arranged by output formulation (rows) and by the prediction horizon of the FCS-MPC teacher controller (columns). The classification networks reproduce the teacher controller's switching decision far more faithfully than the regression networks: the overall accuracy is 92.5% for NNPR1 and 96.8% for NNPR2, against 60.7% for NNRG1 and 74.8% for NNRG2. The two classifiers are strongly diagonally dominant; for NNPR2, the per-state recall of the six active vectors lies between 97.0% and 97.7%, so almost every active-vector decision of the two-step teacher controller is recovered exactly. The regression networks, by contrast, degrade at every switching state, their active-vector recall falling to 58.3–64.3% for NNRG1 and 75.4–86.6% for NNRG2, which indicates that the penalty attached to the regression formulation is not confined to a single class but is distributed across the whole switching-state set.

The structure of the residual errors is itself informative. For the pattern-recognition neural networks, misclassification of an active vector is confined to its two hexagonally adjacent active vectors and to the null state (class 7), with effectively no leakage to the non-adjacent active vectors. In NNPR2, for example, the samples whose true state is the first active vector are assigned only to the two neighbouring active vectors and to the null state, never to the three non-adjacent active vectors. Such confusions are benign for the control task, because adjacent active vectors differ by the smallest step in the $\alpha\beta$ plane and therefore impress almost the same voltage on the LC filter. The regression networks do not preserve this locality. The reason for this could be the following: because each output is fitted independently under a squared-error criterion and the applied state is then taken by the argmax, the decision does not enforce the mutual exclusivity that the softmax imposes, and thus errors spread more widely across the switching-state set, reaching non-adjacent vectors whose associated voltage error is larger.

The merged null state—namely, class 7 (column 7 in Figure 6)—is the hardest class for every network, and its behaviour under the classifier (panels (a) and (b) of Figure 6) isolates the effect of the training horizon. The one-step teacher controller applies a null vector in only 1.50% of samples, so NNPR1 encounters this class rarely and learns it poorly: its null-state recall is only 9.2%, and the few true null-state samples are dispersed almost uniformly across the six active outputs. The two-step teacher controller exercises the null vector far more often, in 8.12% of samples, and the identical architecture then recovers a null-state recall of 91.5% in NNPR2. The deficit is therefore one of representation rather than of the classifier itself, and it is removed once the richer two-step policy populates the class. This is the under-representation effect reported by Wang et al. [26], reproduced here because of the prediction horizon rather than the converter topology, and it provides direct, sample-level evidence for the hypothesis that the two-step teacher controller transfers a measurable advantage to its imitator.

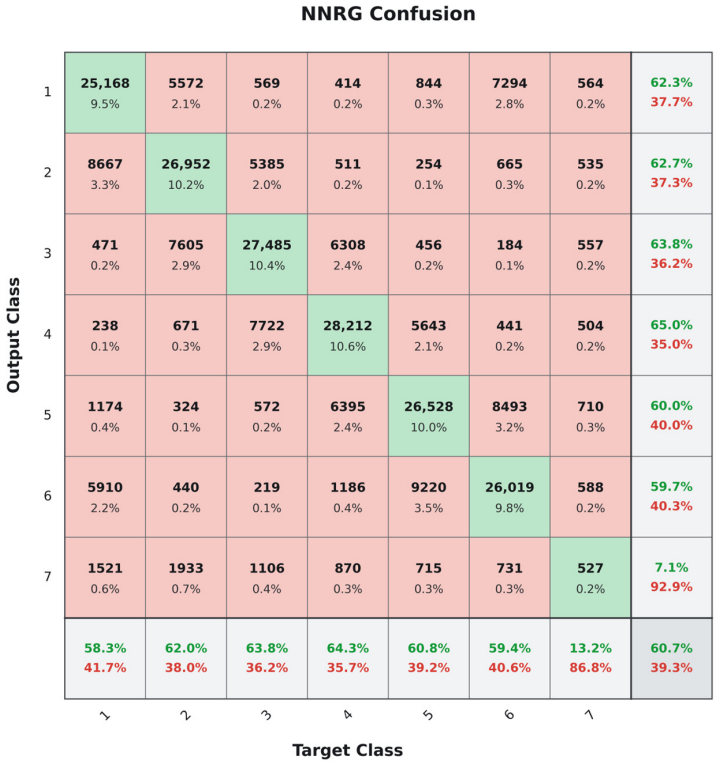


(**a**)

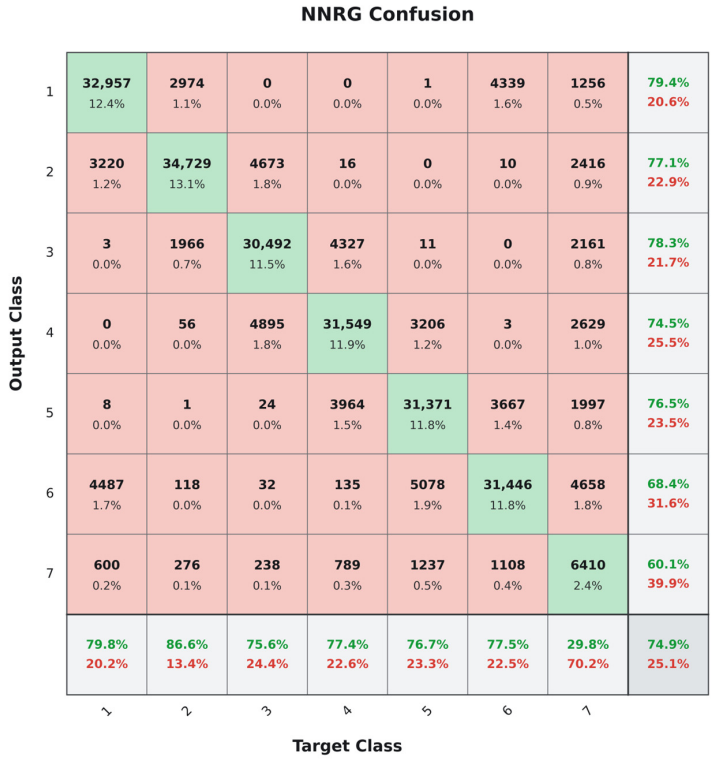


(**b**)

Figure 6. *Cont.*



(**c**)



(**d**)

Figure 6. Confusion matrices of the four neural controllers, evaluated on the complete dataset of each teacher controller (265,072 samples, including the training, validation, and test samples). (**a**) NNPR1, the pattern-recognition network trained on the one-step FCS-MPC teacher controller. (**b**) NNPR2, the pattern-recognition network trained on the two-step teacher controller. (**c**) NNRG1, the regression network trained on the one-step teacher controller. (**d**) NNRG2, the regression network trained on the two-step teacher controller.

Because the null state is under-represented in the one-step data, overall accuracy can hide poor performance in that class. We therefore also calculated class-balanced measures from the confusion matrices in Figure 6. The balanced accuracy (the mean of the seven per-class recall values) is 81.7% for NNPR1, 96.5% for NNPR2, 54.5% for NNRG1, and 71.8% for NNRG2, and the macro-averaged F1 score is 82.2%, 96.4%, 54.3%, and 71.8%, respectively. For NNPR1, the gap between overall accuracy (92.5%) and balanced accuracy (81.7%) comes almost entirely from the null state, whose F1 score is 16.7%; for NNPR2, the two measures differ by only 0.3 percentage points. Training used no class weighting or resampling, so these values reflect the class distribution produced by each teacher controller.

That the classification networks dominate the regression networks here agrees with the simulation results of Wang et al. [26] for the modular multilevel converter, in which the pattern-recognition controller gave a lower circulating current than the regression controller (0.73 A against 0.81 A RMS), but not with their experiments, in which the regression controller performed better. This discrepancy reflects the structure of their classifier rather than an ill-posed comparison. In that work, the classifier was realised as a bank of independent per-feature binary classifiers, in which insert-numbers that occurred rarely were learned poorly, so the joint regressor performed better on hardware. The classifier adopted here is instead a single seven-class softmax defined over the complete switching-state set, whose competitive normalisation sharpens the argmax decision rather than fragmenting it. A seven-output classifier of this kind, trained on one-hot targets, was also used by Mohamed et al. for one-step imitation on the same inverter topology, where it gave lower THD than its FCS-MPC teacher controller (0.72% against 1.92%) [23]. Provided each class is adequately represented, the classification network therefore reproduced the decisions of its teacher controller more accurately than the regression network at every switching state in this study, and the sole case in which it did not—the under-populated null state of NNPR1—is a property of the one-step training data that is corrected by the two-step teacher controller.

3.5. Deployment

Once trained, each network is reduced to a fixed set of numeric parameters for deployment. The input and output weight matrices, the two bias vectors, and the stored normalisation parameters are written as numeric constants into a MATLAB Function block. At each sampling instant the block assembles the eight-dimensional $\alpha\beta$ input from the measured signals, applies the stored normalisation, performs one pass through the hidden layer and the output layer, and selects the switching state—by the largest softmax probability for NNPR, or the largest linear output for NNRG—which is then applied to the inverter. The online computation is thus a single forward pass of fixed size: its cost does not depend on the prediction horizon of the FCS-MPC controller that generated the training data, because the benefit of the longer horizon is embedded in the network weights during training rather than recomputed online. This fixed, horizon-independent cost is the property examined quantitatively in the execution-time results of Section 4.

4. Simulation Results

4.1. Simulation Setup

The proposed NNRG and NNPR controllers were both trained on one- and two-step FCS-MPC data generated in MATLAB/Simulink (2023b), applied to the inverter with output LC filter shown in Figure 1 with balanced linear resistive loads only. Nonlinear and unbalanced loads were used neither in training nor in the tests of Sections 4.2–4.6, only in the robustness tests of Section 4.8. Simulations and training were conducted on an Apple M3 MacBook with 16 GB unified memory running macOS Sequoia 15.6.1.

One- and two-step FCS-MPC, along with both NNRG and NNPR controllers, were used to control the inverter system in Figure 1 under the single case presented in Table 1 and also under the training conditions. Both neural networks were benchmarked against their classical FCS-MPC counterpart. Performance was also compared across training data type—one-step NNRG versus one-step NNPR, and two-step NNRG versus two-step NNPR—to evaluate classification against regression as control strategies.

4.2. Steady-State Performance

Figure 7 compares the steady-state capacitor voltage V_c, load current i_o, and phase-A of the filter current i_f for the two FCS-MPC strategies and the four neural controllers trained on them, under the operating condition presented in Table 1. The load current is near-identical across all six controllers, confirming that every formulation tracks the reference and delivers a clean output to the load; the discrimination therefore lies in the voltage ripple and in the high-frequency content of i_f, against which the per-row THD is reported.

At this operating point the two-step FCS-MPC panel (1.15%) shows lower distortion than the one-step panel (3.11%), in line with the general expectation that a longer horizon improves switching quality; the coarser one-step switching shows as the heavier hash on i_f in (a) relative to (b). The one-step NNPR (3.16%) reproduces its teacher controller almost exactly, in both THD and the visible ripple of (a) and (c), which is the expected outcome of training the network to imitate that strategy. The two-step NNPR panel (1.01%) sits slightly below its two-step teacher controller on this single run; we treat this apparent improvement cautiously at the panel level and return to it across all sixteen conditions later in the section. NNRG is the weaker neural controller under both teacher controllers—3.69% against NNPR's 3.16% on one-step data and 1.40% against 1.01% on two-step—though on two-step data the gap is small: panels (f) and (d) are visually comparable, and the degradation that is plain in the noisier one-step (e) waveform has largely closed.

Table 2 shows the output-voltage THD for all six controllers at each of the sixteen operating conditions used for training, spanning load resistances from 5 Ω to 1000 Ω and three DC-link and reference-voltage pairs. The two FCS-MPC controllers serve as baselines, and each neural network is evaluated against the horizon of the controller whose decisions it was trained to reproduce.

Averaged over the sixteen conditions, the NNPR controller trained on two-step data gives the lowest mean THD at 0.89%, followed by the two-step NNRG at 1.36%, the two-step FCS-MPC baseline at 1.73%, the one-step FCS-MPC baseline at 2.32%, the one-step NNPR at 2.55%, and the one-step NNRG at 2.89%. The ordering separates the controllers into a clear high-quality group trained on two-step data and a lower-quality group trained on one-step data, with the prediction horizon of the training data acting as the dominant factor.

The 2 × 2 ablation in Figure 8 makes this comparison clearer. Moving from one-step to two-step training data lowers the mean THD from 2.55% to 0.89% for the classification network (NNPR) and from 2.89% to 1.36% for the regression network (NNRG); averaged across the two formulations, the richer two-step data reduces mean THD by 58%, from 2.72% to 1.13%. The choice of NN type has a smaller but consistent secondary effect: classification outperforms regression in both regimes, by 0.34 percentage points with one-step data and by 0.47 percentage points with two-step data. The two effects are largely separable—the horizon term dominates and the type term adds to it, with only a weak interaction (the classification advantage grows by 0.13 percentage points on two-step data). On this evidence, the best combination is the classification formulation trained on two-step data; the margin attributable to the formulation alone is small enough, however, that it is better read as a consistent secondary effect rather than as a decisive one.

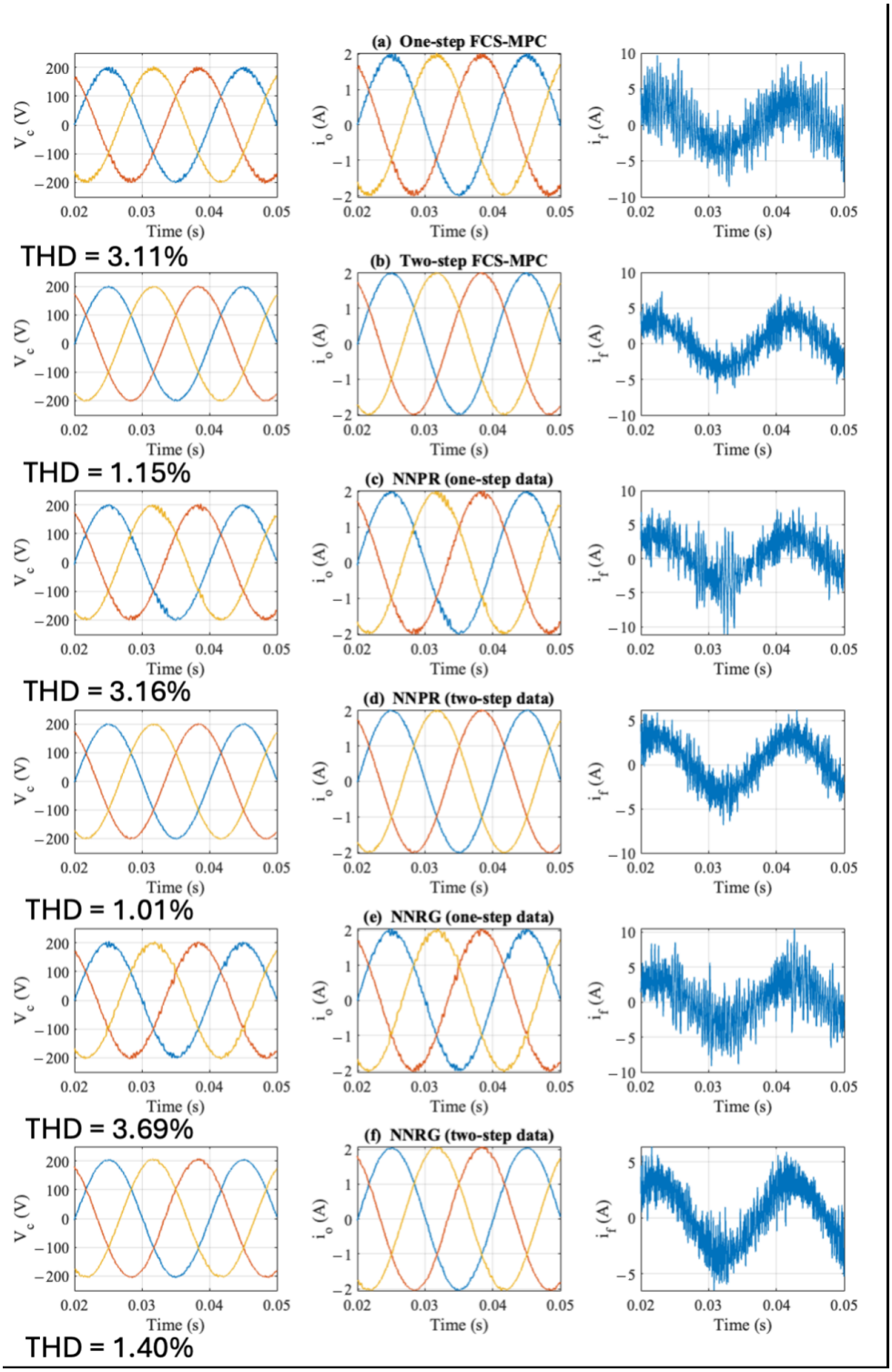


Figure 7. Waveform comparison of all controllers under the conditions in Table 1. Both types of neural networks were trained on one- and two-step FCS-MPC strategies. The annotated THD values are computed for phase A of the capacitor voltage V_c. (**a**) One-step FCS-MPC; (**b**) two-step FCS-MPC; (**c**) NNPR trained on one-step data; (**d**) NNPR trained on two-step data; (**e**) NNRG trained on one-step data; (**f**) NNRG trained on two-step data. The four NN-based controllers are MATLAB Function blocks based on the neural networks' weights and biases. Phase A is blue, phase B is orange, and phase C is yellow. Each phase is separated by 120°.

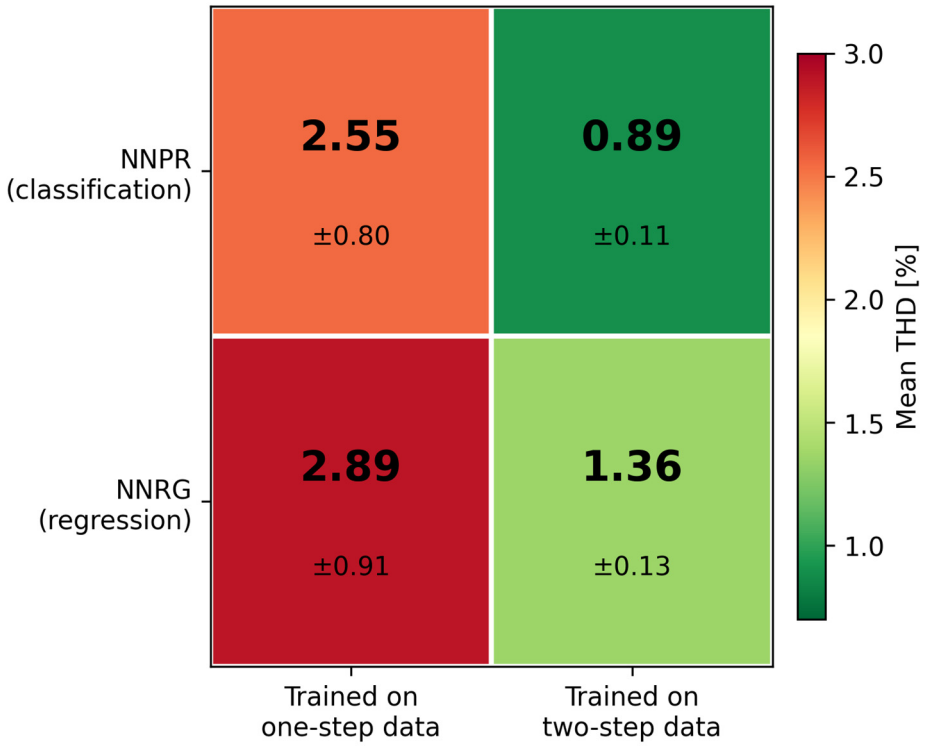


Figure 8. Mean output-voltage THD of the four NN controllers across the sixteen operating conditions, isolating the 2 × 2 ablation of NN controller (NNPR classification vs. NNRG regression) against training-data FCS-MPC horizon (one-step vs. two-step). Cell annotations give mean ± standard deviation in %; lower (greener) is better.

Paired comparisons over the sixteen operating conditions of Table 2 were made with the exact two-sided Wilcoxon signed-rank test, which does not assume normally distributed differences. NNPR2 gave lower THD than both two-step FCS-MPC (mean paired difference −0.84 percentage points, $p < 0.001$) and NNRG2 (−0.47 points, $p < 0.001$) at all sixteen conditions. Moving from one-step to two-step training data reduced THD at all sixteen conditions for NNPR (−1.66 points, $p < 0.001$) and at fifteen for NNRG (−1.53 points, $p < 0.001$). On one-step data, the classifier's advantage was smaller and less consistent: NNPR1 was below NNRG1 at twelve conditions (−0.34 points, $p = 0.029$) and above one-step FCS-MPC at thirteen (+0.23 points, $p = 0.025$). After Holm correction for the six tests, the first four comparisons remain at $p < 0.001$ and the two one-step comparisons give $p = 0.0499$. Because each condition is a single deterministic simulation, these tests show that the differences are consistent across the operating range, not that they would persist under measurement noise or on hardware.

Beyond the mean values, and setting aside for the moment that the two-step NNPR improves on the two-step FCS-MPC teacher controller it was trained to imitate—reducing mean THD from 1.73% to 0.89%, a result examined in Sections 4.7 and 5.2—the spread across operating conditions is itself informative, shown as the error bars in Figure 9. The two-step NNPR is not only the lowest controller but by far the most consistent, with a standard deviation of 0.11% over the sixteen conditions, against 0.22% for the two-step FCS-MPC baseline and 0.58% for the one-step FCS-MPC baseline. The FCS-MPC baselines follow the expected physical trend, with THD rising as the load resistance increases and the LC filter becomes less damped: the one-step baseline climbs from 1.24% at 5 Ω to 3.18% at 1000 Ω within the baseline group. The two-step NNPR remains essentially flat across the same range, which indicates that it reproduces high-quality switching uniformly across the load, DC-link, and reference variations covered by the sixteen conditions, rather than

only at isolated points. This consistency should not be over-read: on its own it speaks to the uniformity of the learned control law, not to the NNPR-versus-NNRG comparison, which rests on the mean and spread figures reported above. A further caveat applies to all sixteen points—they are the conditions on which the networks were trained, so the table shows that NNPR has learned the control law well across the training envelope rather than that it generalises beyond it; the generalisation question is taken up separately at the unseen interpolated points in Section 4.6.

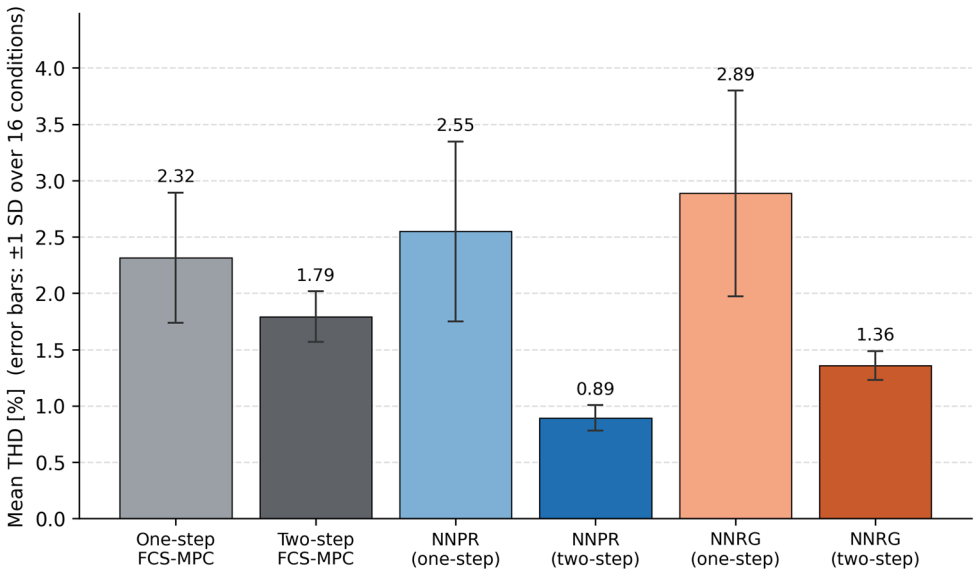


Figure 9. Mean output-voltage THD for the six controllers across the sixteen operating conditions, with error bars showing ±1 standard deviation. The two-step NNPR achieves both the lowest mean (0.89%) and the smallest spread, while controllers trained on one-step data show markedly wider variation.

4.3. Transient-State Performance

Figure 10 shows the transient response of the capacitor-voltage magnitude envelope, |Vc|, for all six controllers following a startup from zero to the reference voltage v_c^*. The simulations use the baseline operating point presented in Table 1. Plotting the αβ magnitude rather than the three individual phase voltages collapses the response onto a single envelope, so the rise, overshoot, and settling of every controller can be read on one axis.

The settling time is the last instant at which the envelope leaves this band; a controller whose envelope never remains inside the band does not settle within the window. Alongside the settling time, the RMSE deviation of the envelope from the 200 V reference over the full 0–5 ms window quantifies how closely each controller holds the reference across the whole transient window rather than at the single settling instant. Because every controller shares the same initial 0 to 200 V rise, these RMSE values are large and dominated by that common ramp; they are meaningful relative to one another, not against the steady-state tracking error reported later.

The clearest pattern is that the prediction horizon, more than the controller type, governs transient quality. The three two-step controllers form a low-error group (RMSE 42.93–50.02 V, settling time 0.84–2.13 ms), while the three one-step controllers form a high-error group (RMSE 60.74–65.78 V, settling time 4.26–4.86 ms). This grouping carries one qualification: each neural controller is trained to imitate an FCS-MPC controller of a given horizon, so it is falling into the same group as the one that the teacher controller is partly built in, and the only fully independent horizon comparison available here is one-step against two-step FCS-MPC. With that caveat, the separation is consistent and sizeable. The one-step controllers all remain under-damped, plausibly because with no anticipation of the

step belonging to time instant $k + 1$, each over-corrects and oscillates across the band for the full window, and the differences between them are minor against this shared behaviour.

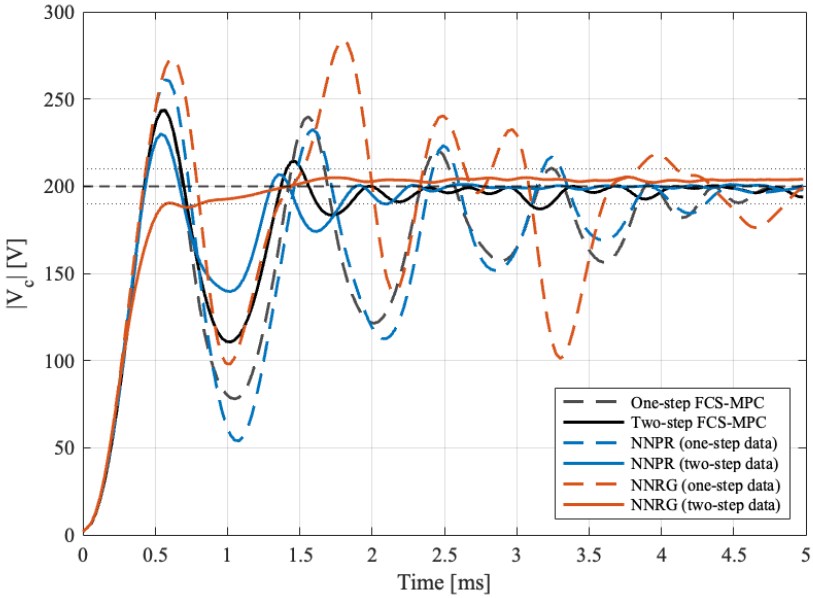


Figure 10. Transient response of the capacitor-voltage magnitude envelope |Vc| for all six controllers during start-up from zero to the 200 V reference, under the baseline operating conditions of Table 1. The dashed horizontal line marks the 200 V reference and the two dotted lines mark the ±5% settling band (190–210 V).

Within the two-step group, the three controllers separate by how faithfully they reproduce the FCS-MPC control law. Two-step FCS-MPC is the well-damped baseline, settling in 1.86 ms with an envelope RMSE of 50.02 V. NNPR, trained on the two-step data, tracks this baseline closely (settling time 2.13 ms) and holds the reference slightly tighter across the transient window (RMSE = 45.95 V, a modest reduction of roughly 8% on the baseline). This is consistent with the cause identified in Section 4.7: the network receives the measured load current, whereas its teacher controller relies on a noisy estimate of it. The margin is small enough that little should rest on this single transient; it gains weight only in combination with the THD and load-step results, which are explored in the next section.

NNRG, trained on the same two-step data, behaves differently. It settles fastest of all six controllers (0.84 ms), rises to the reference without overshoot, and records the lowest envelope RMSE (42.93 V). On a startup transient both metrics reward precisely this behaviour—a smooth, overshoot-free rise that parks at the reference—so on these two numbers alone NNRG appears strongest. We read this with caution, because the same response is also what an over-damped, less aggressive controller would produce, and a startup alone cannot distinguish the two cases. The likely mechanism is the regression objective itself: under a mean-squared-error loss the network predicts the average optimal action in each region of the input space, and because the two-step labels contain frequent anticipatory low-magnitude (braking) vectors, that average is biased toward gentler actions and the closed loop is detuned. The accompanying small steady-state offset, visible as the envelope settling slightly above 200 V, is consistent with a regressor that minimises average error rather than enforcing exact tracking, although a positive offset is not an inevitable consequence of regression and may also reflect the asymmetry of the discrete action set or the rectifying effect of working with a magnitude envelope. Nonetheless, the low settling time and RMSE here are at least as readily explained by reduced effective control authority as by better tracking, so the verdict on NNRG is deliberately reserved for the load-step and THD results prior to this section.

The one-step group confirms that this over-damping is specific to the regression-and-two-step pairing. One-step FCS-MPC oscillates heavily and settles only at 4.26 ms (RMSE = 62.04 V), and NNPR trained on one-step data reproduces this under-damped law almost exactly (settling time 4.32 ms, RMSE = 65.78 V). NNRG trained on one-step data remains equally aggressive (settling time 4.86 ms, RMSE = 60.74 V): the one-step labels contain no anticipatory braking for the regression to over-generalise, so the averaging that detunes the two-step network has nothing to bias, and NNRG stays under-damped. The contrast between NNRG's response on one-step and two-step data is the clearest evidence available here that its damped behaviour is a property of the training data rather than of the regression network itself. However, this remains an inference from the observed pattern instead of an actual demonstration.

Taken together, the transient results indicate that horizon length is the primary driver of damping across the controllers tested; that NNPR closely reproduces its FCS-MPC teacher controller under both horizons and slightly improves on the two-step teacher controller, consistent with the load-current effect of Section 4.7; and that NNRG departs from the teacher controller only when trained on two-step data, where it over-damps the response. Whether this over-damping is an asset or a liability cannot be judged from a startup transient alone and is settled by the load step response explored in the next section.

4.4. Load-Step Response

Figures 11 and 12 present the load-step response of all six controllers. Figure 11 reports the RMS tracking error of the output voltage against its 200 V reference in a sliding 1 ms window, and Figure 12 reports the mean deviation of the voltage magnitude from the same reference. Each figure is split by prediction horizon so that the one-step and two-step controllers are read on common axes. At t = 50 ms a second identical resistive load is switched in parallel with the first, halving the load resistance from 100 Ω to 50 Ω and doubling the load current from 2 A to 4 A while the reference is held fixed; the operating point is the baseline of Table 1. Because the load current is fixed by the load rather than commanded, the disturbance is instantaneous and identical in every run, so differences in the response are attributable to the controller. The one caveat is that the controllers enter the step from slightly different steady states, which the offset column of Table 3 makes explicit.

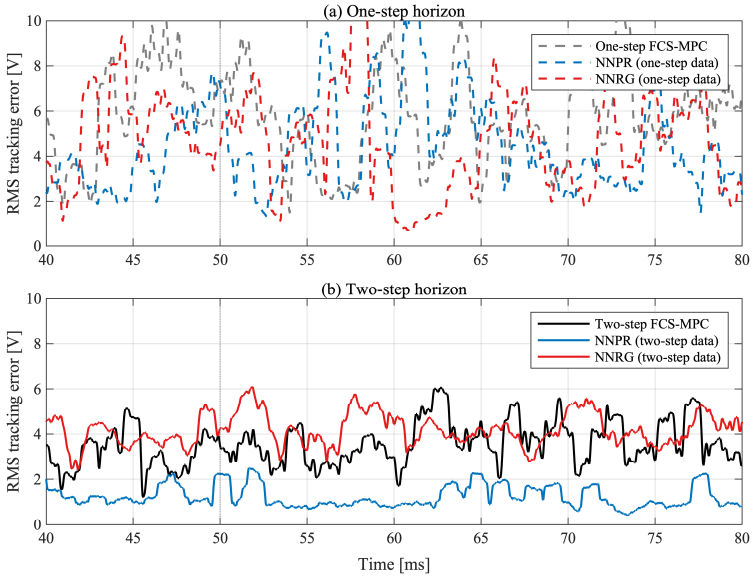


Figure 11. RMS tracking error of the output voltage through the load step, for the one-step (**a**) and two-step (**b**) controllers, evaluated in a sliding 1 ms window.

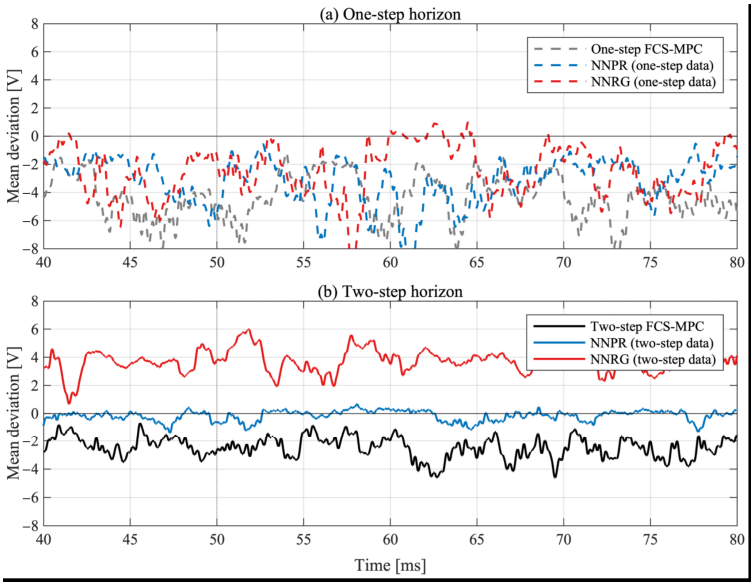


Figure 12. Mean deviation of the output-voltage magnitude from the 200 V reference through the load step, for the one-step (**a**) and two-step (**b**) controllers.

Table 3. Load-step response metrics: voltage sag, recovery time, sustained RMS tracking error, and steady-state offset for each controller following a 100 to 50 Ω load step at t = 50 ms. "Does not settle" denotes controllers that ripple across the ±5% band continuously; "within band" denotes controllers that remain inside it throughout.

Controller	Sag [V]	Recovery [ms]	RMSE [V]	SS Offset [V]
One-step FCS-MPC	15.19	does not settle	6.16	−4.32
Two-step FCS-MPC	7.74	43.4	4.13	−2.75
NNPR (one-step)	9.70	does not settle	4.88	−3.05
NNPR (two-step)	5.78	within band	1.32	−0.19
NNRG (one-step)	13.10	does not settle	5.43	−2.37
NNRG (two-step)	0.03	within band	4.20	+3.76

The output voltage is barely disturbed by the load doubling. The filter capacitor holds the bus stiff and each controller takes up the extra load current within one or two samples, so the sag is brief and small next to the steady-state ripple: it ranges from 0.03 V to 15.19 V across the six controllers (Table 3) and lasts well under a millisecond. The step is therefore rejected almost entirely at the voltage level, which is the behaviour wanted from a UPS-type inverter. What separates the controllers is not the transient but the sustained tracking error around it, and that is what Figures 11 and 12 show.

As at start-up, the prediction horizon sets the dominant pattern. In the one-step panels of both figures the three controllers are indistinguishable within their own ripple: sustained errors sit between 4.88 V and 6.16 V, none re-settle cleanly inside the ±5% band, and all keep crossing it as they did in steady state, so no recovery time can be defined. Whether the network is a classifier or a regressor makes no visible difference at one step. The two-step controllers track far more tightly, and the two-step neural controllers in particular stay inside the ±5% band for the whole step with no recovery transient. The gap between the one-step and two-step groups is wider than any gap within either group, and that is the central pattern of the load-step data.

Within the two-step group the controllers separate clearly. Two-step FCS-MPC holds a sustained error of 4.13 V. NNPR, trained on the two-step data, tracks to 1.32 V, the lowest of any controller and roughly a third of its own teacher controller's error, while its mean deviation sits almost on the reference at −0.19 V (Figure 12). Read at face value, the imitator

rejects the disturbance more accurately than the FCS-MPC controller it was trained to copy. Section 4.7, however, shows that the two-step teacher controller is held back by its estimate of the load current: with the measured load current, its steady-state output amplitude at this operating point rises from 197.3 V to 199.3 V, against 199.7 V for NNPR2 and a 200 V reference. The tighter tracking of NNPR2 is therefore consistent with the same cause as its lower THD (the network receives the measured load current, whereas its teacher controller relied on an estimate) rather than with a better solution of the control problem. We did not repeat the load step with the measured load current.

NNRG, trained on the same two-step data, behaves differently. Its sag is the shallowest of the six at 0.03 V, which on the sag column alone would rank it first for disturbance rejection. It is not. Figure 12 shows why: NNRG sits about 3.8 V above the reference throughout, the only one of the six with a positive offset (+3.76 V, Table 3). Because it already sits above 200 V, the downward pull of the load step moves it briefly toward the reference rather than away from it, so almost no sag registers. The metric is rewarding a standing bias, not a fast recovery. Once the step settles the bias remains, and on the measures that track accuracy, NNRG is the weakest of the two-step group: a sustained RMSE of 4.20 V, level with its teacher controller, and a +3.76 V offset worse than every other controller. Every controller except NNRG settles slightly below the reference, by 4.32 V (one-step FCS-MPC), 2.75 V (two-step FCS-MPC) and just 0.19 V (two-step NNPR), which is the droop expected of a finite-horizon controller without integral action under heavier load. The sign reversal is the discriminating fact: NNRG's error is not load droop but a fixed positive bias, largely independent of load, which is what a regressor that minimises average error rather than enforcing exact tracking will produce. The interpretation rests on the sign, which is robust, not on the precise mechanism; it is the same offset seen on the start-up envelope, now quantified and shown to persist under load.

Taken with the transient results, the load step shows that horizon length governs disturbance rejection across the controllers tested, that the two-step NNPR reproduces and modestly improves on its FCS-MPC teacher controller on sustained tracking, mainly because the teacher controller relies on an estimated load current (Section 4.7), and that the regression formulation buys its over-damped, low-sag character at the cost of a fixed positive bias unique among the six. The bus is held stiff by all of them, so the separating factor is accuracy, and on the two measures that reflect it—sustained RMSE and steady-state offset—the two-step NNPR gives the lowest values of the six controllers at this operating point, which is consistent with the load-current effect of Section 4.7 rather than evidence of a better control law. This is read as a specific limitation of the two-step regressor rather than a failure of the controller as a whole, and it holds at the single baseline operating point of Table 1; behaviour away from that point is examined in Section 4.6.

4.5. Execution-Time Comparison

The execution times of the four neural controllers and the two FCS-MPC baselines were determined by timing the function block computation; the results are reported in Table 4.

Three observations follow from the data. First, the computational cost of the neural controllers is independent of the prediction horizon: all four networks lie within the 28–31 ms range irrespective of whether they were trained on one-step or two-step data, since the deployed architecture (8, 15, and 7 neurons in each layer) is identical in every case and the horizon is encoded in the training data rather than in the network; thus, the richer control policy is embedded into the networks without the added cost. FCS-MPC exhibits the opposite behaviour, its cost doubling from 52 ms to 104 ms as the horizon is extended from one step to two. Second, the two-step NNPR executes in 28 ms relative to the 104 ms required by the two-step FCS-MPC controller on which it was trained, corresponding to a 3.7×

reduction in execution time (73%) while also giving lower mean THD (0.89% against 1.73%, Table 2) and lower sustained RMSE through the load step (1.32 V against 4.13 V, Table 3). In this simulation, therefore, the network did not trade control accuracy for speed. This should not be read as a general accuracy advantage over FCS-MPC: Section 4.7 attributes most of the THD difference to the load-current signal rather than to the network, and the horizon cost is relocated to a one-time training stage rather than removed, as set out below. Third, the classification and regression formulations are computationally indistinguishable, with NNPR (28–31 ms) and NNRG (29–30 ms) incurring comparable near-fixed inference times, such that the selection between them may be governed by control performance alone, with no associated computational penalty.

Table 4. Comparison of the execution time of all six controllers. The displayed execution time is the total time it took to execute the given MATLAB function block in SIMULINK.

Controller	Execution Time [ms]
One-step NNPR	31
Two-step NNPR	28
One-step NNRG	30
Two-step NNRG	29
One-step FCS-MPC	52
Two-step FCS-MPC	104

These figures do not, however, constitute a complete account of the comparison, and three qualifications must be attached to them. Firstly, the runtime saving is incurred once, a priori, in training and data generation rather than at inference. Each network was trained on data produced by executing the corresponding FCS-MPC controller across the sixteen operating conditions, and the two-step networks inherit the cost of generating two-step MPC labels—the same 104 ms computation time that their runtime subsequently circumvents. The 28 ms figure thus represents the cost of deployment rather than the total cost. Secondly, the network forgoes the explicit constraint handling that constitutes a principal motivation for FCS-MPC: the MPC cost function evaluates each candidate switching vector against the system constraints at every step and is guaranteed to return a feasible solution, whereas the network reproduces only the input-to-output mapping presented during training, with no online verification that the selected vector is non-prohibitive. Thirdly, the fixed 8, 15, and 7 neuron architecture, which is precisely the source of the network's horizon-independent cost, also imposes a ceiling: the horizon advantage is fixed at training time, and the network cannot elect at runtime to extend its prediction in the manner available to FCS-MPC through the enlargement of its search. Its behaviour beyond the training envelope consequently rests on interpolation—evidenced here by the four unseen interpolated operating points, which are explored in Section 4.6—rather than on online optimisation.

Finally, a qualification that concerns the environment measurement must be made clear. The reported timings were obtained in MATLAB, whose vectorized linear algebra favors the network's dense matrix operations to a greater degree than the sequential, scalar cost-function evaluation of FCS-MPC; the measured interval should therefore be interpreted as indicative rather than as a precise performance ratio. The more robust claim is the hardware-level one: network inference comprises a fixed sequence of multiply–accumulate operations that maps efficiently onto embedded fixed-point and parallel architectures, whereas the FCS-MPC search scales as 7^N with the horizon and constitutes the binding real-time constraint. On balance, the execution-time results support the central argument—that the network based on pattern recognition secures the accuracy of two-step prediction at a lower, fixed cost that is independent of horizon—provided the figures are interpreted as a deployment

cost under the previously stated training and constraint assumptions. Nevertheless, one must note that the cost of operating the networks online is still lower than the online cost of both FCS-MPC strategies. However, ML algorithms' cost effectiveness—and their performance relative to FCS-MPC—cannot be proven definitively without deployment on embedded hardware.

Because the MATLAB timings depend on the simulation environment, we also give the online computation as an operation count, which does not. Per sample, each network evaluates $8 \times 15 + 15 \times 7 = 225$ multiply–accumulate operations, 15 hyperbolic-tangent evaluations, and a comparison of seven outputs; the softmax need not be evaluated online because it does not change which output is largest. Exhaustive one-step FCS-MPC evaluates 7 state predictions with (15) and 7 cost functions (19), and exhaustive two-step FCS-MPC evaluates $7 + 49 = 56$ state predictions and 49 cost functions (20). The network count is fixed by its size, whereas the FCS-MPC count grows as 7^N. The time per operation depends on the processor and number format and will be measured in the hardware study that follows this work (Section 6).

4.6. Generalisation to Unseen Conditions

In this section, the performance of the four neural network controllers is compared under operating conditions that are not present in the training data to evaluate their generalisation ability. The four test cases interpolate the trained grid in load resistance, DC-link voltage and reference, while the filter parameters remain unchanged. The total harmonic distortion obtained at each condition is reported in Table 5.

Table 5. THD comparison of neural controllers under unseen conditions. Fixed parameters: $Ts = 30$ µs, $f = 50$ Hz, $C = 40$ µF, $L = 2.5$ mH.

Test	R [Ω]	V_{dc} [V]	v_c^* [V]	NNPR (Two-Step) THD [%]	NNRG (Two-Step) THD [%]	NNPR (One-Step) THD [%]	NNRG (One-Step) THD [%]
U1	40	500	200	0.97	1.33	2.15	2.57
U2	150	500	200	0.87	1.40	3.61	3.39
U3	50	550	225	0.87	1.26	1.89	3.01
U4	10	450	175	0.69	1.24	1.54	1.96

The ranking observed under the training conditions is preserved at every unseen point. The NNPR controller trained on the two-step teacher controller achieves the lowest THD in all four cases, ranging from 0.69% to 0.97% with a mean of 0.85%. This range lies within the 0.69–1.01% band recorded across the sixteen training conditions, so no measurable increase in distortion is observed at the interpolated points. The two-step NNPR therefore kept its performance at these four interior points; four points do not establish generalisation across the whole operating envelope (Section 5.3).

Two factors distinguish the controllers, and their effects combine. The prediction horizon of the teacher controller is the dominant factor: for both neural networks and at every test point, the controller trained on the two-step teacher controller produces a lower THD than its one-step counterpart. The mean THD increases from 0.85% to 2.30% for NNPR and from 1.31% to 2.73% for NNRG when the one-step teacher controller is used. The network type is a smaller but consistent factor: the classifier (NNPR) outperforms the regressor (NNRG) in all four cases under the two-step teacher controller and in three of four under the one-step teacher controller. The best controller combines both advantages, and the worst pairs the weaker teacher controller with the regression formulation.

This can be attributed to the fact that the two-step teacher controller selects each switching state by accounting for the inverter dynamics over two sampling intervals, which produces a more consistent labelling of the inputs than the one-step FCS-MPC strategy. The network inherits this labelling implicitly through its training data. Within either teacher controller, the classifier is the more robust of the two formulations because it learns discrete boundaries between input regions and switching vectors, and in doing so filters the label noise introduced by the estimated load current of the teacher controller (Section 4.7). At an unseen interior point, interpolated within the training data, the sinusoidal trajectory falls within regions the classifier has already partitioned, so its discrete decision transfers without degradation. The regressor maps its inputs continuously and propagates small input deviations into its output, which results in higher distortion. This interpretation is consistent with the present data rather than established by it.

The condition U2 (R = 150 Ω) is the only exception to the general pattern and does not contradict it. This resistance interpolates the trained points at 100 Ω and 200 Ω, a high-resistance, light-load regime in which the one-step controllers already performed poorly under the training conditions. The elevated THD of the one-step controllers at U2 (3.61% and 3.39%) reflects the inherited weakness of the one-step teacher controller in this regime rather than a failure specific to the unseen point. The two-step controllers, which performed well across the same regime during training, remained accurate at U2 (0.87% and 1.40%). The minor reversal of the NNPR and NNRG order at this point falls within the already-degraded one-step band and is not interpreted further.

4.7. Switching Frequency and Agreement with the Teacher Controller

The THD comparisons above do not account for switching frequency, which can change THD regardless of how closely a network reproduces its teacher controller [22]. We therefore calculated the average device switching frequency of each controller from the switching states logged at every sampling instant over the last fifteen fundamental cycles of each run:

$$f_{\mathrm{sw}} = \frac{n_{\mathrm{a}} + n_{\mathrm{b}} + n_{\mathrm{c}}}{6T_{\mathrm{w}}} \tag{24}$$

where f_{sw} is the average switching frequency of one power device; n_{a}, n_{b}, and n_{c} are the numbers of state changes of legs a, b, and c, that is, the numbers of sampling instants at which S_{a}, S_{b}, or S_{c} (Section 2.1) differs from its value at the preceding instant; and T_{w} is the length of the counting window, here 0.3 s (fifteen fundamental cycles, or 10,000 sampling periods). Each switching cycle of a device contains one turn-on and one turn-off, so a leg that changes state n_{x} times in T_{w} completes $n_{\mathrm{x}}/2$ switching cycles and switches at $n_{\mathrm{x}}/(2T_{\mathrm{w}})$. Averaging this over the three legs gives the factor of 6 in the denominator.

We first give fsw for the switching states actually applied. Because a hardware implementation would apply whichever zero vector needs fewer transitions, we also give fsw with each null state counted as the zero vector, (0,0,0) or (1,1,1), that needs fewer leg transitions from the preceding state; this common rule is used for all frequency comparisons below. Table 6 lists both quantities, with the THD of the same runs, at the baseline condition of Table 1 and as means over the sixteen conditions of Table 2.

For this section, we repeated the runs with the same model and settings as in Table 2. The mean THD of every controller reproduces Table 2 to within 0.1 percentage points, but the values at individual conditions differ; at the baseline condition, the largest difference is 0.73 points, for two-step FCS-MPC. A closed-loop FCS-MPC trajectory is sensitive to small numerical differences, and a one-cycle THD window samples a single cycle of it, so all comparisons in this section are made within this set of runs.

Table 6. Average switching frequency f_{sw} and output-voltage THD of the six controllers and of two-step FCS-MPC with the measured load current, at the baseline condition of Table 1 and as means over the sixteen conditions of Table 2 (mean ± one standard deviation for f_{sw}). The last f_{sw} column counts each null state as the zero vector that needs fewer transitions. THD values are from the same runs.

Controller	f_{sw}, Baseline [kHz]	f_{sw}, Mean [kHz]	f_{sw}, Mean, Common Rule [kHz]	THD, Baseline [%]	THD, Mean [%]
One-step FCS-MPC	5.86	6.26 ± 0.66	6.21	2.62	2.31
Two-step FCS-MPC	7.80	7.89 ± 0.39	7.76	1.88	1.75
Two-step FCS-MPC, measured i_o	5.80	5.91 ± 0.29	5.55	1.11	1.02
One-step NNPR	4.29	4.76 ± 0.72	4.75	2.79	2.45
Two-step NNPR	5.75	5.83 ± 0.20	5.23	0.97	0.88
One-step NNRG	5.36	5.67 ± 0.45	5.67	3.38	2.99
Two-step NNRG	7.04	7.06 ± 0.22	6.68	1.35	1.35

NNPR2 switches less often than its teacher controller at every condition: its mean f_{sw} is 5.83 kHz against 7.89 kHz for two-step FCS-MPC, a ratio of 0.74 (0.68–0.80 over the sixteen conditions), or 5.23 against 7.76 kHz under the common zero-vector rule. Each of the other three networks also switches less often than its own teacher controller (Table 6). The effect identified by Novak and Dragičević [22]—a higher switching frequency in the imitator, therefore, cannot explain the lower THD of NNPR2.

To compare NNPR2 and its teacher controller at equal switching frequency, the two-step cost (20) is extended with a switching term:

$$J_{2,\lambda} = J_2 + \lambda_{sw} n_{sw} \tag{25}$$

$$n_{sw} = \sum_{x \in \{a,b,c\}} \left(|S_x(k) - S_x(k-1)| + |S_x(k+1) - S_x(k)| \right) \tag{26}$$

where J_2 is the two-step cost (20); $J_{2,\lambda}$ is the cost minimised by the modified controller; $\lambda_{sw} \geq 0$ is a weighting factor, in V^2 per transition because J_2 is a sum of squared voltage errors; and n_{sw} is the number of leg transitions that the candidate two-step sequence requires. In (26), $S_x(k-1)$ is the state, 0 or 1, of leg x applied in the previous sampling period, and $S_x(k)$ and $S_x(k+1)$ are its states in the first and second vectors of the candidate sequence. Each absolute value equals 1 if the leg changes state and 0 if it does not, so n_{sw} counts the transitions over both steps and lies between 0 and 6. When a candidate vector is the null state, it is taken as whichever of (0,0,0) and (1,1,1) needs fewer transitions. With $\lambda_{sw} = 0$, (25) reduces to (20). Increasing λ_{sw} makes sequences with fewer transitions cheaper relative to sequences with a smaller tracking error, which lowers the switching frequency at the cost of a larger predicted error.

At $\lambda_{sw} = 0$, the modified controller reproduces the two-step teacher controller exactly; at the baseline condition, its capacitor-voltage trajectory is identical to that of the original controller at all 16,667 samples. We ran the teacher controller at the baseline condition with $\lambda_{sw} = 0$, 2, 5, 10, 20, 40, and 80 (Figure 13). Its THD first falls from 1.88% at $\lambda_{sw} = 0$ to 1.46% at $\lambda_{sw} = 2$, although f_{sw} falls by only 6%. THD then stays between 1.46% and 1.52% down to 4.72 kHz and rises steeply at lower frequencies. Interpolated to the 5.20 kHz of NNPR2, the teacher controller gives 1.47%, against 0.97% for NNPR2, so matching the switching frequency narrows the difference at this condition from 0.91 to 0.50 percentage points but does not close it. The initial fall in THD shows that some of the switching of

the original controller does not improve the output voltage; the cause is identified below. Because NNPR2 already switches less often than its teacher controller, frequency matching only requires lowering the switching frequency of the teacher controller, so the network was not retrained at each value of λ_{sw}.

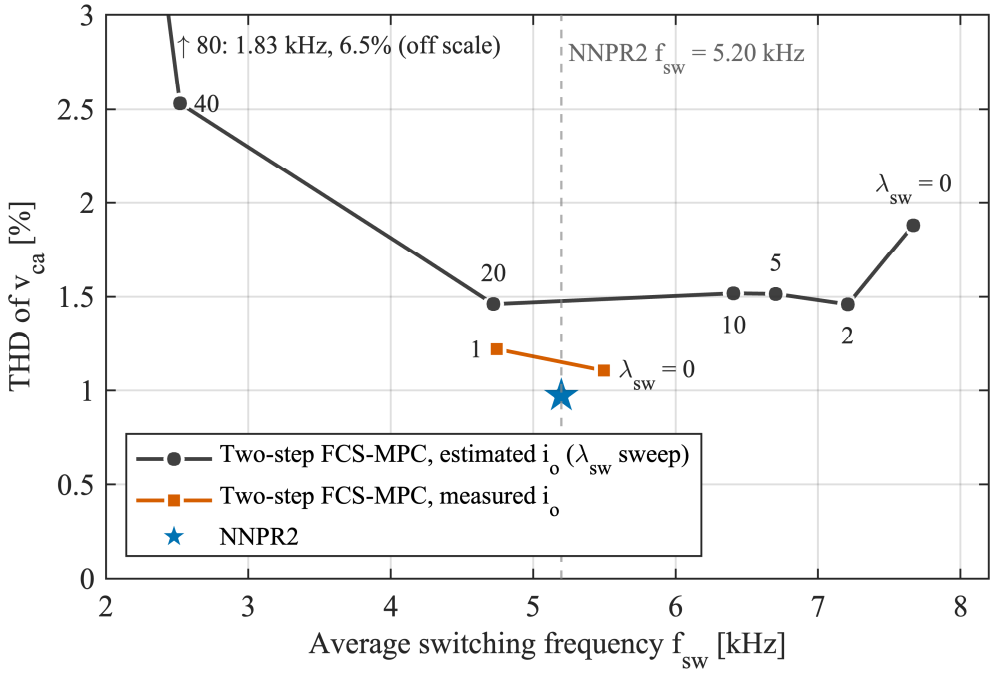


Figure 13. Output-voltage THD against average switching frequency at the baseline condition of Table 1. Two-step FCS-MPC with the estimated load current (the teacher controller of NNPR2) is shown for λ_{sw} = 0–80, two-step FCS-MPC with the measured load current for λ_{sw} = 0 and 1, and NNPR2. f_{sw} is counted with the common zero-vector rule. The dashed line marks f_{sw} of NNPR2. The point at λ_{sw} = 80 (1.83 kHz, 6.51%) lies above the plotted range.

The two-step teacher controller does not measure the load current; it uses the backward-difference estimate (21) and holds it over the horizon, so the estimate carries part of the capacitor-voltage ripple (Section 2.4.2). The networks were trained on this estimate, both as an input and through the labels, but in closed-loop operation they receive the measured load current (Section 3.1). We therefore also ran the teacher controller with the measured load current at all sixteen conditions, holding it over the horizon in the same way and with λ_{sw} = 0 (Table 6). This removes the estimate from the comparison and changes nothing else in the controller. With the measured load current, the mean THD of the teacher controller falls from 1.75% to 1.02% and is lower at all sixteen conditions (exact two-sided Wilcoxon signed-rank test, $p < 0.001$), and its mean f_{sw} falls from 7.89 to 5.91 kHz (from 7.76 to 5.55 kHz under the common rule), close to that of NNPR2. The difference in mean THD between NNPR2 and its teacher controller falls from 0.87 to 0.13 percentage points, so the load-current estimate accounts for 85% of it. The remaining difference is small but consistent: NNPR2 is lower at fifteen of the sixteen conditions ($p < 0.001$). At the baseline condition, interpolation between λ_{sw} = 0 and 1 places this controller at 1.15% at 5.20 kHz, against 0.97% for NNPR2 (Figure 13). The same estimate also accounts for part of the tracking difference in Section 4.4: at the baseline condition, the fundamental amplitude of v_c is 197.3 V for the original controller, 199.3 V with the measured load current, and 199.7 V for NNPR2, against a 200 V reference.

Finally, to test whether NNPR2 obtains its lower THD by minimising (20) more effectively, we evaluated the two-step cost (20) offline along the closed-loop trajectories, using

the load-current estimate (21) of the teacher controller. At every sample, the evaluation records whether NNPR2 selected the J_2-optimal first vector and the excess cost of its choice:

$$\Delta J_2 = \min_j J_2(s_{\mathrm{NN}}, j) - \min_{i,j} J_2(i, j) \tag{27}$$

where i and j index the first and second vectors of a two-step sequence, each taking one of the seven values of Figure 2; $J_2(i, j)$ is the cost (20) of the sequence (i, j) predicted from the state measured at instant k; and s_{NN} is the first vector selected by NNPR2 at that instant. The first term is the lowest cost still reachable once the first vector is fixed to the choice of the network, and the second is the cost of the sequence that two-step FCS-MPC would select, so ΔJ_2 is zero when NNPR2 selects the J_2-optimal first vector and positive otherwise. We report the mean of ΔJ_2 over all samples as a percentage of the mean optimal cost.

As a check, when applied to the runs of the teacher controller itself (fifteen conditions), the evaluation selects the same first vector as the controller at every sample. Along the NNPR2 trajectories, it agrees with the network at 79.9% of samples on average (74.5–82.8% over the thirteen conditions evaluated): 77.1% for active vectors and 94.8% for the null state. Most disagreements involve an adjacent active vector or the null state. The mean excess cost is 8.2% of the mean optimal cost (5.0–14.2%). NNPR2 therefore achieves its lower THD at a higher value of the cost of its teacher controller, not by minimising (20) more effectively.

The lower THD of NNPR2 therefore comes mainly from the load-current signal, not from a higher switching frequency or a better minimisation of (20). The network was trained on labels and inputs that contain the estimate, yet in closed loop it receives the measured load current, which the original teacher controller never used. Given the same measured signal, the teacher controller comes within 0.13 percentage points of the network. We have not retrained the network on the labels of this controller in this work; retraining is planned together with the delay-compensated labels described in Section 6.

4.8. Robustness to Measurement Noise, Delay, Parameter Mismatch, and Load Type

The tests in Sections 4.2–4.6 use noise-free measurements, nominal filter parameters, and balanced linear resistive loads. We therefore also tested the six controllers, and two-step FCS-MPC with the measured load current of Section 4.7, without retraining at the baseline operating point of Table 1. The parameter-mismatch test follows the practice in [28] of evaluating learning-based predictive control against component degradation in simulation. The test conditions are:

- Measurement noise: zero-mean Gaussian noise is added to each phase of the measured v_c, i_f, and i_o before they reach the controller, with standard deviations of 0.5%, 1%, and 2% of the peak value of each signal at the baseline condition. The same noise sequence is used for every controller.
- Computation delay: a one-sample delay separates the calculation of the switching state from its application. No controller compensates for it, and the networks are not retrained. Retraining on delay-compensated labels is the first step of the hardware route outlined in Section 6.
- Parameter mismatch: the plant inductance and capacitance are changed by −20% and +20%, one at a time, while the FCS-MPC prediction model and the training data keep the nominal values of Table 1.
- Nonlinear load: a three-phase diode-bridge rectifier with a DC-side capacitor of 300 µF and a resistor of 60 Ω, as used in [23].
- Unbalanced load: 100 Ω in phase a and 50 Ω in phases b and c, with an isolated star point.
- Extrapolation: V_{dc} = 650 V with v_c^* = 275 V at R = 100 Ω, and R = 2000 Ω at V_{dc} = 500 V and v_c^* = 200 V, both outside the trained ranges of Table 2.

For each run, the THD of each phase voltage is computed over the last fifteen fundamental cycles (0.2–0.5 s), including every non-fundamental component up to the Nyquist frequency, and the highest of the three phase values is reported. We do not use a one-cycle window because it samples a single cycle of a trajectory that is sensitive to small numerical differences (Section 4.7). The steady-state offset is the mean of $|v_c| - v_c^*$ over the same window, as in Table 3. Each controller is compared with its own baseline run, evaluated over the same window; these baseline runs reproduce the runs of Section 4.7 sample for sample. Table 7 marks any controller whose steady-state offset leaves the ±5% band of Section 4.4. A network whose THD rises by more than that of its teacher controller under a given test is treated as less robust to that condition. Because the load-current estimate (21) differentiates the measured capacitor voltage, measurement noise and capacitance mismatch act on the estimate of the teacher controllers as well as on their prediction model, so two-step FCS-MPC with the measured load current serves as a second reference for the two-step networks.

Table 7. Worst-phase output-voltage THD [%] of the seven controllers at the baseline operating point of Table 1 under each robustness test, without retraining. THD is computed over the last fifteen fundamental cycles, so the baseline row differs slightly from the one-cycle values of Tables 2 and 6. MPC1 and MPC2 denote one- and two-step FCS-MPC, and MPC2c denotes two-step FCS-MPC with the measured load current. † marks a steady-state offset outside ±5% of v_c^*.

Test	MPC1	MPC2	MPC2c	NNPR1	NNPR2	NNRG1	NNRG2
Baseline	2.83	1.89	1.17	2.97	0.97	3.93	1.53
Noise 0.5%	4.53	2.63	1.76	4.12	1.27	4.46	1.60
Noise 1%	7.53 †	3.76	2.66	5.17	1.88	5.36	1.77
Noise 2%	11.57 †	6.38 †	4.31	7.93 †	3.39	7.23	2.55
One-sample delay	497.9 †	61.9 †	103.3 †	747.9 †	50.3 †	181.4	5.61
L −20%	3.56	2.48	1.61	3.79	1.13	4.31	1.81
L +20%	2.31	1.58	1.02	2.29	0.87	3.17	1.34
C −20%	4.15	2.30	2.33	11.49 †	1.69	8.35	1.88
C +20%	2.27	1.74	0.80	1.70	0.71	2.31	1.34
V_{dc} 650 V, v_c^* 275 V	2.91	1.91	1.14	2.85	0.91	3.89	1.55
R = 2000 Ω	3.27	1.97	1.28	3.61	1.03	3.80	1.51
Unbalanced load	2.78	1.88	1.16	2.83	0.95	3.37	1.48
Nonlinear load	2.74	2.49	1.86	3.28	1.78	4.37	2.74

Measurement noise raised the THD of every controller, but at all three noise levels every network degraded less than its own teacher controller. At 2% noise, the THD of NNPR2 rose by 2.42 percentage points to 3.39%, against 4.49 points (to 6.38%) for two-step FCS-MPC and 3.14 points (to 4.31%) for two-step FCS-MPC with the measured load current. NNRG2 changed least, rising by 1.02 points to 2.55%. The one-step controllers were the most sensitive: at 2% noise, one-step FCS-MPC reached 11.57% and NNPR1 7.93%, with offsets of −15.3 V and −11.0 V. Two-step FCS-MPC also left the ±5% band, at −12.9 V. Because NNPR2 degraded less than the controller with the measured load current at every noise level, the estimate (21) does not account for all of the smaller degradation of the networks. These values come from a single noise sequence.

Without compensation, a one-sample delay raised the worst-phase THD of six of the seven controllers to between 50% and 748%, reduced their average switching frequency to 0.66–1.19 kHz, and shifted their offsets to between −40 V and +224 V. NNRG2 was least affected, with a THD of 5.61% and an offset of −1.9 V, although its THD still rose by 4.08 points. Because the teacher controllers are affected as strongly as the networks, the robustness criterion is not applied to this test. The result agrees with the known need to compensate the computation delay in FCS-MPC [15] and confirms that, before hardware testing, the labels must be regenerated with that compensation and the networks retrained

(Section 6). In their present form, therefore, none of the controllers, including NNPR2, is suitable for real-time implementation without delay compensation.

A 20% reduction in the filter inductance raised the THD of every controller by 0.17–0.82 points, with the smallest rise for NNPR2, whereas a 20% increase in either the inductance or the capacitance lowered the THD of every controller. A 20% reduction in the capacitance was the most demanding mismatch for the one-step networks: NNPR1 rose by 8.52 points to 11.49%, with an offset of −13.0 V, and NNRG1 by 4.41 points to 8.35%, against 1.32 points for their teacher controller. The two-step networks rose by 0.72 points (NNPR2) and 0.35 points (NNRG2), compared with 0.42 points for two-step FCS-MPC and 1.17 points for the same controller with the measured load current.

At V_{dc} = 650 V and v_c^* = 275 V, the THD of every controller changed by no more than 0.12 points, even though about 55% of the samples presented to the networks had at least one input outside the training range because the reference amplitude exceeds the trained maximum of 250 V. At R = 2000 Ω, the largest rise was 0.63 points (NNPR1), and the two-step networks changed by less than 0.07 points. The unbalanced load did not raise the THD of any controller. The nonlinear load, which no network saw during training, raised the THD by up to 1.21 points (NNRG2). NNPR2 rose by 0.81 points, slightly more than two-step FCS-MPC (0.60) and two-step FCS-MPC with the measured load current (0.69), but its THD of 1.78% remained the lowest of the seven controllers.

Across the twelve tests, NNPR2 gave the lowest THD of the seven controllers in ten and was second, behind NNRG2, at 1% and 2% noise. Its offset left the ±5% band only under the uncompensated delay, and its THD never rose by more than 0.30 points above the rise in its teacher controller. The largest margins against a teacher controller were those of the one-step networks under the 20% capacitance reduction (7.20 points for NNPR1 and 3.09 points for NNRG1). Each value comes from a single deterministic run at the baseline operating point, so the tests show how each controller responds to each condition, not how that response varies across the operating range or between noise sequences.

5. Discussion

5.1. Dominance of the Teacher Controller's Prediction Horizon

The four evaluation criteria of Section 4 agree on a single ordering. The prediction horizon of the FCS-MPC controller that generated the training labels separates the six controllers more strongly than the choice between the classification and the regression formulation, and it does so on every measure: mean steady-state THD falls from 2.72% to 1.13% when averaged across the two network types, the two-step controllers settle in 0.84–2.13 ms against 4.26–4.86 ms for the one-step group, sustained tracking error through the load step separates into a 1.32–4.20 V band and a 4.88–6.16 V band along the same division, and the ordering is preserved without exception at the four unseen interpolated conditions. The network formulation acts as a consistent but secondary factor, favouring the classifier by 0.34 percentage points of mean THD on one-step data and by 0.47 percentage points on two-step data. One qualification attaches to this reading: each neural controller imitates an FCS-MPC controller of a given horizon, so its grouping with that teacher controller is in part built into the experiment, and the only horizon comparison that is fully independent of the imitation step is the one between the two FCS-MPC baselines themselves. Subject to that qualification, the practical implication is that effort is better spent on enriching the teacher policy than on selecting between network formulations of comparable size.

5.2. Why the Two-Step Classifier Gives Lower THD than Its Teacher Controller

In Table 2, the two-step NNPR gives lower THD than the two-step FCS-MPC controller it was trained to imitate (0.89% against 1.73%), whereas the one-step NNPR sits marginally above its own teacher controller (2.55% against 2.32%). An exact imitator could at best equal the controller it copies, so a lower THD must come either from a difference in the information available to the network and its teacher controller in closed loop or from a suboptimal teacher controller. The tests of Section 4.7 identify the main cause. The two-step teacher controller estimates the load current from a backward difference in the capacitor voltage (21), which carries the voltage ripple into the estimate, and the networks were trained on labels and inputs that contain this estimate; in closed-loop operation, however, they receive the measured load current. Given the same measured signal, the mean THD of the teacher controller falls from 1.75% to 1.02% and its mean switching frequency from 7.89 to 5.91 kHz, and the difference between NNPR2 and its teacher controller shrinks by 85%, from 0.87 to 0.13 percentage points of mean THD. The one-step teacher controller uses the same estimate but was not rerun with the measured load current, so we do not know whether the estimate also explains why NNPR1 does not improve on its teacher controller.

The remaining difference of 0.13 percentage points is small but consistent, appearing at fifteen of the sixteen conditions, and the present data do not identify its cause. It is not a switching-frequency effect of the kind reported in [22]. NNPR2 switches less often than its original teacher controller at every condition and, on average, less often than the teacher controller with the measured load current, and at equal switching frequency at the baseline condition it remains 0.18 percentage points below the latter (Figure 13). The network also does not minimise the two-step cost (20) more effectively: it departs from the J_2-optimal first vector at about one sample in five, with a mean excess cost of 8.2% of the optimal cost. Two-step FCS-MPC minimises the predicted tracking error over two sampling periods, not the THD over a fundamental period, so a control law that sometimes departs from the J_2-optimal choice can still give lower THD; the remaining difference may reflect the smoother decision boundaries of the classifier. The simulation environment does not by itself favour one controller, because all controllers run in the same Simulink model with the same solver, sampling period, and measurement path; Section 4.8 tests measurement noise and computation delay, which may affect a learned control law differently. Mohamed et al. also reported lower THD for an imitator than for its teacher controller in one-step imitation on the same topology (0.72% against 1.92% for FCS-MPC) [23]. The present results show that such comparisons should give the imitator and its teacher controller the same measured signals in closed loop.

5.3. Limitations

Five limitations bound the claims made above. Firstly, the NN-based controllers and the FCS-MPC baselines have been evaluated through simulation only; no hardware, hardware-in-the-loop (HIL), or embedded implementation has been performed, and hardware or HIL validation remains necessary to establish the practical applicability of the approach. The reported execution times characterise a MATLAB environment whose vectorised linear algebra favours the network's dense matrix operations over the sequential cost-function evaluation of FCS-MPC, and the measured ratio should be read as indicative rather than as a hardware figure. Secondly, the comparison between NNPR2 and its teacher controller depends on the load-current signal each receives: the networks were trained on the estimated load current used by the teacher controllers but receive the measured load current in closed loop, which accounts for most of the difference in THD (Section 4.7). We did not retrain the networks on labels generated with the measured load current or rerun the one-step teacher controller with it. Thirdly, the network reproduces only the input-to-output mapping

presented during training and performs no online feasibility check, so the explicit constraint handling that motivates FCS-MPC is lost at deployment. Keeping the optimiser online and using learning only to improve its predictions, as in [28], retains constraint handling but also keeps the online optimisation cost. Fourthly, no stability guarantee is offered for the learned control law; the evidence for well-behaved closed-loop operation is empirical, drawn from the transient and load-step responses of Sections 4.3 and 4.4 and the robustness tests of Section 4.8, and a formal analysis of the imitator's stability margins remains outstanding. The results therefore do not show that the networks are more stable than FCS-MPC, only that no instability was observed at the simulated conditions, apart from the large deviations under an uncompensated computation delay. Fifthly, the robustness tests of Section 4.8 show significant performance degradation under computation delay. With an uncompensated one-sample delay between the calculation of the switching state and its application, the worst-phase THD of six of the seven controllers, including NNPR2 (from 0.97% to 50.3%) and both FCS-MPC teacher controllers, rose above 50%, and their steady-state offsets left the ±5% band; NNRG2 was least affected, but its THD still rose from 1.53% to 5.61%. The FCS-MPC formulation of Section 2 does not compensate for this delay, so the training labels assume that the selected vector acts at the instant at which the states are sampled. Because a one-sample delay is unavoidable in a digital implementation, neither the networks nor their teacher controllers can be implemented in real time in their present form. Appropriate delay compensation, such as that of [15], must be included in the teacher controllers, and the networks retrained on the resulting labels, before real-time implementation, as in the experimentally validated imitator of [22].

It is noted that all four conditions are interpolations within the trained ranges of resistance, DC-link voltage and reference voltage; the LC filter hardware is unchanged; and four points are a limited sample. The results show robustness to interpolation within the trained operating envelope. Section 4.8 extends this evidence to two extrapolated operating points, ±20% filter-parameter mismatch, measurement noise, and unbalanced and nonlinear loads, but only at the baseline operating point, with one noise sequence, and without hardware effects such as quantisation and dead time. Those tests also show that an uncompensated one-sample computation delay severely degrades both the FCS-MPC controllers and the networks. All training data, and the tests of Sections 4.2–4.6, use balanced linear resistive loads.

6. Conclusions and Future Study

This work compared neural network imitators of FCS-MPC against the FCS-MPC controllers they were trained to reproduce, for a two-level three-phase voltage-source inverter with an output LC filter. Four imitators were formed by crossing two network types, a pattern-recognition classifier (NNPR) and a regression network (NNRG), with one-step and two-step teacher controller data, and all were evaluated through MATLAB/Simulink simulations only, with balanced linear resistive loads across sixteen operating conditions and at unseen interpolated points. The horizon of the training data mattered more than the choice of network, and the classifier gave lower mean THD at both horizons, although on one-step data its advantage was small and not consistent across conditions (NNPR1 below NNRG1 at twelve of the sixteen conditions). The two-step NNPR gave the lowest mean output-voltage THD, 0.89%, below the 1.73% of the two-step FCS-MPC teacher controller it imitates; kept a similar THD (0.69–0.97%) at four unseen conditions inside the training envelope; and runs at the fixed cost of a single network forward pass. These comparisons come from deterministic simulations at the tested conditions; they show how the controllers compare in that setting, not that the networks outperform FCS-MPC in general.

A lower THD than the teacher controller does not mean that the imitator optimises beyond it. NNPR2 switches less often than its teacher controller and departs from the J_2-optimal choice in about one sample in five, so its advantage comes from neither a higher switching frequency nor a better solution to the two-step cost. It comes mainly from the load-current signal: the teacher controller estimates the load current from a backward difference that carries the voltage ripple, whereas the network, although trained on this estimate, receives the measured load current in closed loop. Given the measured load current, the teacher controller comes within 0.13 percentage points of the mean THD of the network. Imitators and their teacher controllers should therefore be compared with the same measured signals, and the quality of a learned controller depends on the signals its teacher controller uses as well as on its horizon. All results were obtained through simulation only. Hardware or hardware-in-the-loop (HIL) validation is therefore required to establish the practical applicability of the NN-based controllers and to confirm their performance under measurement noise, timing constraints, and parameter drift before the approach can be recommended for deployment. The robustness tests also showed significant performance degradation under computation delay: an uncompensated one-sample delay raised the THD of NNPR2 from 0.97% to 50.3% and that of both FCS-MPC teacher controllers above 60% (Section 4.8). Appropriate delay compensation must therefore be included in the teacher controllers, and the networks retrained, before real-time implementation. No stability guarantee is given for the learned control law: its closed-loop behaviour has been shown empirically, in simulation and at the conditions tested, but not proven.

Four directions follow from this work. The first is hardware-in-the-loop and subsequent embedded validation, which is required both to confirm the control-quality ranking under measurement noise and timing constraints and to replace the indicative MATLAB execution times with figures obtained on the target processor. The second is to retrain the networks on labels generated with the measured load current and with the delay compensation of [15], so that the remaining 0.13-point difference between NNPR2 and its teacher controller can be examined. The third is the extension of the evaluation beyond interpolation to supply and reference voltages outside the trained ranges, to nonlinear and unbalanced loads, and to the filter parameter drift introduced by component ageing. The fourth is a stability analysis of the learned control law, which the present empirical evidence does not supply. The robustness tests of Section 4.8 address the third direction at one operating point in simulation and show that the computation delay must be compensated in the training data before hardware testing. For the first direction, we will regenerate the labels with the delay compensation of [15] and retrain the networks. The trained network, which needs 225 multiply–accumulate operations per sample, can then be generated as C or HDL code from its MATLAB Function block, verified in fixed point against the floating-point network, and run on a processor or FPGA against the same Simulink plant model on a real-time HIL simulator.

Author Contributions: Conceptualisation, A.A.Z. and Y.G.; methodology, A.A.Z. and Y.G.; software, A.A.Z.; validation, A.A.Z. and Y.G.; formal analysis, A.A.Z.; investigation, A.A.Z.; data curation, A.A.Z.; writing—original draft preparation, A.A.Z.; writing—review and editing, A.A.Z. and Y.G.; visualisation, A.A.Z. and Y.G.; supervision, Y.G.; project administration, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Royal Society International Exchanges 2025 Global Round 1 under the Grant Number IES\R1\251429.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: To ensure clarity and precision in the presentation of our findings, the language model Claude Opus 5.5 was used as an assistive tool for drafting and editing the English language of this manuscript and ChatGPT 6 was used for enhancing the quality of some figures. The experimental design, data analysis, and design and execution of simulation were conceived and carried out entirely by the human authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cortés, P.; Ortiz, G.; Yuz, J.I.; Rodríguez, J.; Vazquez, S.; Franquelo, L.G. Model predictive control of an inverter with output LC filter for UPS applications. *IEEE Trans. Ind. Electron.* **2009**, *56*, 1875–1883. [CrossRef]
2. Liu, Z.; Liu, J.; Zhao, Y. A unified control strategy for three-phase inverter in distributed generation. *IEEE Trans. Power Electron.* **2014**, *29*, 1176–1191. [CrossRef]
3. Shen, Y.; Ghaderloo, R.A.; Singhabahu, C.; Resalayyan, R.; Khaligh, A. Design and modeling of a two-stage PV inverter for single phase and three phase applications. In Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE), Nashville, TN, USA, 29 October–2 November 2023; pp. 2326–2332. [CrossRef]
4. Bibave, R.; Thokal, P.; Hajare, R.; Deulkar, A.; William, P.; Chandan, A.T. A comparative analysis of single phase to three phase power converter for input current THD reduction. In Proceedings of the International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 16–18 March 2022; pp. 325–330. [CrossRef]
5. Aslam, A.W.; Hassan, J.; Minambres-Marcos, V.; Al-salloomee, A.G.S.; Roncero-Clemente, C. Traditional and hybrid topologies for single-/three-phase transformerless multilevel inverters. *Electronics* **2024**, *13*, 4058. [CrossRef]
6. Uddin, M.H.; Abid, A.; Inam, U.; Urooj, A.; Siddiqui, M.R. A comparative analysis of a multilevel inverter topology based on phase disposition sinusoidal pulse width modulation. *Eng. Proc.* **2023**, *46*, 30. [CrossRef]
7. Ahmed, B.M.A.M.; Alqamoudi, F.M.F.A.; Abdi, A.F.; Ali, M.I.; Ponniran, A.; Abu Bakar, A. Performance evaluation of three-phase multilevel inverter with R-load using SPWM. *J. Electron. Volt. Appl.* **2026**, *7*, 12–18. [CrossRef]
8. Dhanvi, S.; Govindaraju, H.V. Performance analysis and harmonic mitigation of a grid-connected three-phase UPS inverter using SPWM and LC filter in MATLAB/Simulink. In Proceedings of the Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS), Pathum Thani, Thailand, 25–27 June 2026; pp. 364–369. [CrossRef]
9. Li, P.; Tong, X.; Wang, Z.; Xu, M.; Zhu, J. Sensorless model predictive control of single-phase inverter for UPS applications via accurate load current estimation. *Sensors* **2023**, *23*, 3742. [CrossRef]
10. Kazmierkowski, M.P.; Malesani, L. Current control techniques for three-phase voltage-source PWM converters: A survey. *IEEE Trans. Ind. Electron.* **1998**, *45*, 691–703. [CrossRef]
11. Batkhishig, B.; Gonçalves, P.F.d.C.; Pietrini, G.; Nahid-Mobarakeh, B.; Emadi, A. PWM techniques for two-level voltage source inverters: A comparative study. *IEEE Access* **2025**, *13*, 86235–86255. [CrossRef]
12. Mohamed, I.S.; Zaid, S.A.; Abu-Elyazeed, M.F.; Elsayed, H.M. Classical methods and model predictive control of three-phase inverter with output LC filter for UPS applications. In Proceedings of the International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia, 6–8 May 2013; pp. 483–488. [CrossRef]
13. Harbi, I.; Rodriguez, J.; Liegmann, E.; Makhamreh, H.; Heldwein, M.L.; Novak, M.; Rossi, M.; Abdelrahem, M.; Trabelsi, M.; Ahmed, M.; et al. Model predictive control of multilevel inverters: Challenges, recent advances, and trends. *IEEE Trans. Power Electron.* **2023**, *38*, 10845–10868. [CrossRef]
14. Xia, C.; Liu, T.; Shi, T.; Song, Z. A simplified finite-control-set model-predictive control for power converters. *IEEE Trans. Ind. Inform.* **2014**, *10*, 991–1002. [CrossRef]
15. Cortés, P.; Rodríguez, J.; Silva, C.; Flores, A. Delay compensation in model predictive current control of a three-phase inverter. *IEEE Trans. Ind. Electron.* **2012**, *59*, 1323–1325. [CrossRef]
16. Andreu, M.; Rohten, J.; Espinoza, J.; Silva, J.; Pulido, E.; Leon, L. Enhanced model predictive control using state variable feedback for steady-state error cancellation. *Sensors* **2024**, *24*, 5869. [CrossRef]
17. Cortés, P.; Rodríguez, J.; Vazquez, S.; Franquelo, L.G. Predictive control of a three-phase UPS inverter using two steps prediction horizon. In Proceedings of the IEEE International Conference on Industrial Technology (ICIT), Viña del Mar, Chile, 14–17 March 2010; pp. 1283–1288. [CrossRef]
18. Karamanakos, P.; Geyer, T.; Kennel, R. Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems. In Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE), Montreal, QC, Canada, 20–24 September 2015; pp. 334–341. [CrossRef]

19. Zafra, E.; Vazquez, S.; Geyer, T.; Aguilera, R.P.; Franquelo, L.G. Long prediction horizon FCS-MPC for power converters and drives. *IEEE Open J. Ind. Electron. Soc.* **2023**, *4*, 159–175. [CrossRef]
20. Akinwumi, J.O.; Gao, Y.; West, G.T. Optimisation algorithms in multi-step prediction horizon MPC for two-level inverters. In Proceedings of the International Symposium on Electrical, Electronics and Information Engineering (ISEEIE), Leicester, UK, 28–30 August 2024; pp. 115–122. [CrossRef]
21. Akinwumi, J.O.; Gao, Y.; Yuan, X.; Vazquez, S.; Ruiz, H.S. ANN-based online parameter correction for PMSM control using sphere decoding algorithm. *Sensors* **2026**, *26*, 553. [CrossRef]
22. Novak, M.; Dragičević, T. Supervised imitation learning of finite-set model predictive control systems for power electronics. *IEEE Trans. Ind. Electron.* **2021**, *68*, 1717–1723. [CrossRef]
23. Mohamed, I.S.; Rovetta, S.; Do, T.D.; Dragičević, T.; Diab, A.A.Z. A neural-network-based model predictive control of three-phase inverter with an output LC filter. *IEEE Access* **2019**, *7*, 124737–124749. [CrossRef]
24. Gao, Y.; Wang, S.; Dragičević, T.; Wheeler, P.; Zanchetta, P. Artificial intelligence techniques for enhancing the performance of controllers in power converter-based systems—An overview. *IEEE Open J. Ind. Appl.* **2023**, *4*, 366–375. [CrossRef]
25. Wang, D.; Shen, Z.J.; Yin, X.; Tang, S.; Liu, X.; Zhang, C.; Wang, J.; Rodriguez, J.; Norambuena, M. Model predictive control using artificial neural network for power converters. *IEEE Trans. Ind. Electron.* **2022**, *69*, 3689–3699. [CrossRef]
26. Wang, S.; Dragičević, T.; Gao, Y.; Teodorescu, R. Neural network based model predictive controllers for modular multilevel converters. *IEEE Trans. Energy Convers.* **2021**, *36*, 1562–1571. [CrossRef]
27. Yang, X.; Wang, K.; Kim, J.; Park, K.B. Artificial neural network-based FCS-MPC for three-level inverters. *J. Power Electron.* **2022**, *22*, 2158–2165. [CrossRef]
28. Cavus, M.; Jiang, J.; Allahham, A.; Sun, H. Learning-enhanced stochastic model predictive control with trustworthy forecasting for resilient multi-energy microgrid operation. *Appl. Energy* **2026**, *422*, 128321. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.