

Article

A Neural Network-Based Current Controller for a Three-Phase Grid-Connected Voltage-Source Converter

Byron Alejandro Acuña Acurio ¹, Felipe Grijalva ^{2,*} and Juan Carlos Rodríguez ³

¹ School of Electrical Engineering, Universidade Estadual de Campinas, Campinas 13083-852, SP, Brazil; b209428@dac.unicamp.br

² Colegio de Ciencias e Ingenierías “El Politécnico”, Universidad San Francisco de Quito USFQ, Diego de Robles S/N, Quito 170157, Ecuador

³ Data Center and Energy Business Unit at Analog Devices Inc., Wilmington, MA 01887, USA; juancarlos.rodriguez@analog.com

* Correspondence: fgrijalva@usfq.edu.ec

Abstract

This paper presents a compact neural network (NN) current controller for a three-phase, grid-connected voltage-source converter (VSC). The proposed NN replaces the fixed-gain proportional–integral (PI) compensator within a traditional d - q synchronous frame current-control architecture. An eight-dimensional feature vector of present and past system variables predicts incremental modulation updates through a single-hidden-layer multilayer perceptron (MLP) with 16 hidden neurons and only 178 trainable parameters, requiring 712 bytes of parameter storage. Temporal information is introduced externally through one-step error history and recursive modulation-domain accumulation, avoiding internally recurrent layers such as LSTM cells. Trained on 11 million samples from 10,000 PSIM simulations, the NN controller reduces averaged overshoot by 59%, settling time by 94%, integral absolute error by 27%, and steady-state error by 42% compared to the PI teacher. In simulation-based reference-step tests, the NN controller learns a smoother bounded incremental law from an underdamped PI current-loop teacher, reducing the averaged overshoot, settling time, integral absolute error, and steady-state error magnitude. Hardware-in-the-loop and prototype validation remain outside the scope of this work.

Keywords: grid-connected; neural network control; synchronous reference frame; current control; voltage-source converter; multilayer perceptron



Academic Editors: Nicola Lusardi and Pedro J. Villegas

Received: 25 August 2026

Revised: 25 September 2026

Accepted: 30 September 2026

Published: 3 October 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](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Grid-connected voltage-source converters (VSCs) are the main interface between renewable energy sources, energy-storage systems, and the utility grid [1]. They are used in photovoltaic systems, wind-energy converters, battery systems, and microgrids [2]. In three-phase grid-connected conversion systems, the d - q synchronous frame control is widely used to produce Pulse-Width Modulation (PWM) signals to command the power switches [3]. A standard grid-following implementation includes the functions of grid synchronization, Clarke–Park transformations, and an inner feedback current controller as the primary-level control of the VSC (Figure 1). Thanks to the d - q frame simplification [4], a proportional–integral (PI) controller can be used to command DC variables (i_d , i_q) that are proportional to the three-phase peak currents on the AC side [5]. In practical implementations, grid-voltage feed-forward and cross-coupling compensation [6] are typically added to the current control

loop. In addition, an outer voltage loop can be added to regulate the DC-link voltage or exchange active/reactive power as a secondary-level control [7].

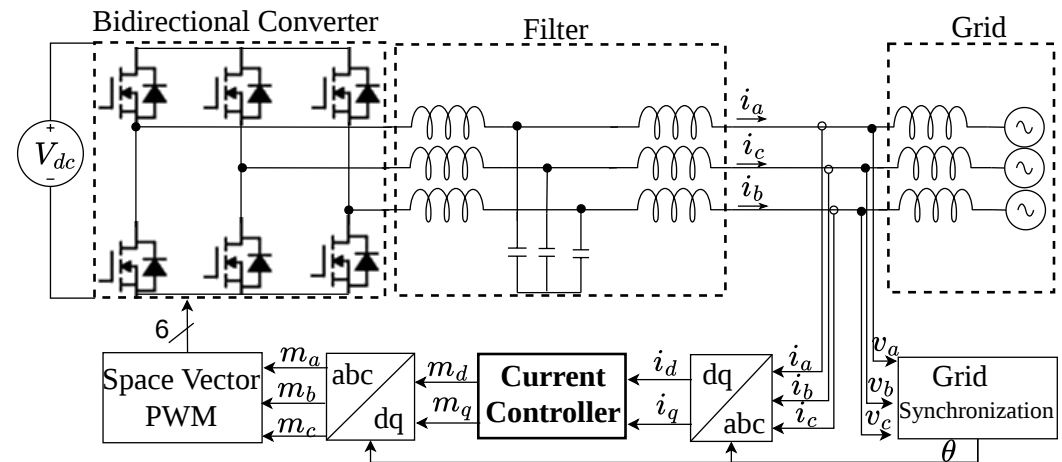


Figure 1. Three-phase grid-connected voltage-source converter.

Current-loop design for grid-connected VSCs has traditionally relied on model-based control (MBC) approaches. These methods use analytical descriptions of the converter, filter, and grid interface, typically expressed in state-space or synchronous-reference-frame form [6]. However, maintaining an accurate plant description can become difficult when the effective dynamics change because of operating-point variation, filter-parameter uncertainty, digital delays, grid-impedance variation, or unmodeled nonlinearities. These limitations motivate data-driven control (DDC) approaches, which can learn control-relevant input–output relationships from simulated or measured data [8].

In standard grid-following implementations, PLL-based grid synchronization provides the reference angle used by the Clarke–Park transformations [9]. In this work, the synchronization stage is kept unchanged in its standard form. The proposed data-driven component is restricted to the inner dq axis current compensator, where it generates modulation index updates.

In the conventional synchronous-reference-frame architecture, the inner dq -axis current loops are commonly implemented with fixed-gain PI compensators. PI controllers remain attractive because they are simple, computationally inexpensive, and effective when tuned around a nominal operating point [10]. However, because their gains are fixed after tuning, their transient response may degrade when the operating point, DC-link voltage, filter parameters, reference commands, or grid-side impedance deviate from the nominal design conditions [11]. Under these conditions, the current loop may exhibit larger overshoot, longer settling time, residual tracking error, and stronger dq -axis coupling.

Neural networks (NNs) provide a compact way to approximate nonlinear control laws in power-electronic converters [12]. Early ANN-based converter controllers demonstrated that learned mappings can improve operation under disturbances and switching nonlinearities [13], while more recent data-driven approaches have been combined with model-predictive control, adaptive PI structures, or recurrent architectures [11,14,15]. These approaches improve flexibility, but they often introduce a larger online optimization problem, an adaptive supervisor, or recurrent memory states that increase implementation complexity. For grid-connected VSCs, the LSTM controller reported in [15] introduces temporal information through internal recurrent cells. That design is useful because a purely memoryless feed-forward map $x[k] \mapsto y[k]$, constructed only from instantaneous variables, does not explicitly encode the dynamic state of the closed-loop converter.

This study addresses a narrower but technically distinct problem: how to retain causal memory in the inner current compensator without using internally recurrent neural network layers. Rather than relying on LSTM or GRU cells, the proposed architecture uses a compact feed-forward multilayer perceptron (MLP), while temporal information is supplied externally through previous current errors, measured currents, previous dq -frame modulation commands, and recursive accumulation of the predicted modulation increments. The network therefore learns an incremental modulation update embedded in a dynamic controller structure, not an isolated static current-to-voltage map.

Table 1 summarizes how the proposed controller is positioned with respect to representative current-control alternatives. The purpose of this comparison is not to claim universal superiority, but to clarify the gap addressed in this paper: a small, supervised, modulation-domain NN current controller that preserves explicit recurrence outside the neural model.

Table 1. Positioning of the proposed controller with respect to related current-control approaches.

Approach	Main Strength	Implementation Burden	Position of This Work
Fixed-gain PI and multivariable PI [16–18]	Simple, deterministic, and widely used for dq -axis current regulation	Requires gain selection and anti-windup design around a nominal model	Learns from an under-damped PI response and recovers a smoother bounded incremental update
Adaptive PI and hybrid ANN-PI [11]	Adjusts effective gains when operating conditions change	Requires an adaptive layer and additional tuning variables	Avoids online gain adaptation; all learned parameters are fixed after training
Predictive and ANN-assisted predictive control [14,19,20]	Handles constraints and multivariable objectives explicitly	May require online cost evaluation, prediction models, or larger computational budgets	Uses a single deterministic forward pass without online optimization
Robust and resonant grid-current controllers [21–23]	Improve operation under grid uncertainty, resonances, or distorted conditions	Depend on a selected plant or impedance model and robust design margins	Does not replace these methods; robustness beyond the nominal balanced grid is stated as a boundary
Recurrent NN current control [15]	Learns temporal behavior through internal recurrent states	More parameters and recurrent-state handling are required	Replaces internal recurrence with external controller variables and modulation-domain accumulation

The main contributions of this research are as follows:

- A modulation-domain neural current controller that predicts bounded incremental dq -axis modulation updates and obtains memory through external recurrence rather than through internal recurrent layers.
- An event-synchronized supervised-learning formulation in which auxiliary integral-error states are used only to identify digital controller update instants, while the deployed controller receives only present errors, one-step error history, measured currents, and previous modulation commands.
- A transparent simulation-based evaluation against the under-damped PI teacher, including statistical transient metrics, an explanation of the learned recovery mechanism, a practical boundedness argument, and a deployment-cost analysis.

The rest of the paper is organized as follows: Section 2 introduces the converter model and the under-damped PI current-loop teacher used for data generation. Section 3 describes the proposed neural network controller. Section 4 explains the data-generation process, training procedure, operating envelope, and performance metrics. Section 5 presents the comparative results and their limitations. Section 6 concludes the paper.

2. Traditional Three-Phase Converter Control

The current controller block in Figure 1 is expanded in Figure 2. A standard grid-following implementation uses a phase-locked loop (PLL) to estimate the synchronous reference-frame angle θ , applies Clarke–Park transformations to the measured voltages and currents, and regulates the resulting d - and q -axis currents with feedback compensators [24]. The outputs of the dq -axis PI compensators are combined with grid-voltage feed-forward and cross-coupling compensation terms, normalized into modulation commands, transformed back to the three-phase frame, and applied to a space-vector PWM stage [25]. In the present work, only the inner current compensator is replaced; the PLL, transformations, feed-forward terms, and modulation stage remain unchanged.

Under the balanced-grid assumption and with the grid-side filter represented by L_f and R_f , the relationship between converter terminal voltage and grid voltage is

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_f \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + L_f \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} v_{ga} \\ v_{gb} \\ v_{gc} \end{bmatrix} \quad (1)$$

where v_a , v_b , and v_c denote the converter terminal voltages. Applying the Clarke–Park transformation gives the synchronous-frame model in Equation (2), where $\omega = 2\pi f$ is the grid angular frequency.

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = L_f \frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + R_f \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} v_{gd} \\ v_{gq} \end{bmatrix} + \omega L_f \begin{bmatrix} -i_q \\ +i_d \end{bmatrix} \quad (2)$$

The PI current-loop teacher used for data generation and comparison follows Equations (3) and (4), where K_p and K_i are the proportional and integral constants:

$$v_d = L_f \frac{di_d}{dt} + R_f i_d = \left(K_p + \frac{K_i}{s} \right) (i_d^{\text{ref}} - i_d) \quad (3)$$

$$v_q = L_f \frac{di_q}{dt} + R_f i_q = \left(K_p + \frac{K_i}{s} \right) (i_q^{\text{ref}} - i_q) \quad (4)$$

The compensator outputs are combined with the grid-voltage feed-forward and decoupling terms shown in Figure 2 and normalized to obtain the commanded modulation indices

$$m_d = \frac{v_d + v_{gd} - \omega L_f i_q}{V_{\text{base}}}, \quad m_q = \frac{v_q + v_{gq} + \omega L_f i_d}{V_{\text{base}}}, \quad (5)$$

where V_{base} is the voltage base of the SVM modulation stage. The PI implementation used as the baseline includes output limiting and anti-windup handling, as is standard in digital current-loop libraries [26]. The same surrounding modulation limits are retained when the PI compensator is replaced by the proposed NN current controller.

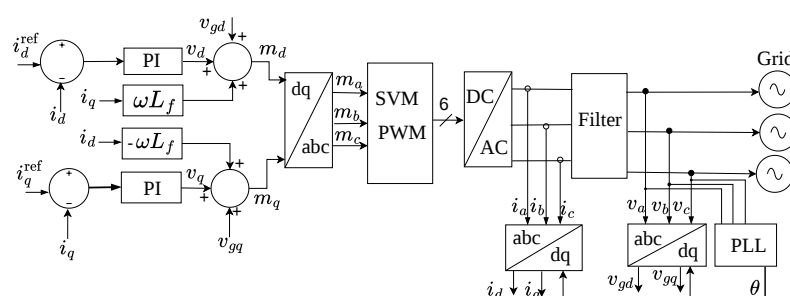


Figure 2. General current-control structure in the dq frame.

3. Proposed Neural Network Controller

The proposed controller replaces the PI current loop with a compact neural network. In the architecture, the PLL, direct and inverse Clarke/Park transformations, and Pulse-Width Modulation (PWM) stages are preserved. The proposed design isolates the current compensator. Hence, the proposed neural network controller maintains seamless compatibility with standard vector-control architectures.

In the remainder of this section, the neural-controller output is denoted by $\mathbf{m}[k] = [m_d[k], m_q[k]]^\top$, where m_d and m_q are the accumulated d - q axis modulation commands applied to the modulation stage. The variables v_d and v_q are reserved for the voltage-domain outputs of the conventional PI compensators.

The practical implementation of the PI block occurs within a digital controller, which requires a discretized model of Equations (3) and (4). Thus, $v_d[k]$ and $v_q[k]$ denote the discrete-time control actions of the d - q axis current compensators, i.e., the outputs of the PI blocks of Figure 2, which enter the modulation-index synthesis of Equation (5). Let T_s denote the sampling period and k the discrete-time index. The standard position form of the d -axis PI controller is expressed as [27]:

$$v_d[k] = K_p e_d[k] + K_i T_s \sum_{n=0}^k e_d[n], \quad (6)$$

where $e_d[k] = i_d^{\text{ref}}[k] - i_d[k]$ is the current-tracking error.

Rather than implementing the position form (6) directly, the proposed approach adopts its *velocity* (incremental) form. Defining the control increment as $\Delta v_d[k] \triangleq v_d[k] - v_d[k-1]$, the update depends only on the present and previous current-tracking errors, $e_d[k]$ and $e_d[k-1]$, as

$$\Delta v_d[k] = K_p (e_d[k] - e_d[k-1]) + K_i T_s e_d[k], \quad (7)$$

an identical expression is obtained for the q -axis with $e_q[k] = i_q^{\text{ref}}[k] - i_q[k]$.

The velocity form contains no explicit integrator state; the integral action is recovered by accumulating the increments, $v_d[k] = v_d[k-1] + \Delta v_d[k]$. In this work, learning is carried out directly at the level of the complete modulation commands $m_d[k]$ and $m_q[k]$ of Equation (5), which are the signals recorded from the simulated converter. Defining the modulation increments $\Delta m_{d,q}[k] \triangleq m_{d,q}[k] - m_{d,q}[k-1]$, the feed-forward terms in (5) vary slowly between consecutive controller updates, so these increments are dominated by the compensator increments of (7), while the residual decoupling contribution is learned from data.

Both the PI controller and the proposed neural controller possess memory. In the PI controller, memory is carried by the integral state. In the proposed controller, memory is encoded externally through the previous errors and previous modulation commands included in (8), coupled with an external accumulation in (11) and (12).

$$\mathbf{x}[k] = \begin{bmatrix} e_d[k] \\ e_q[k] \\ e_d[k-1] \\ e_q[k-1] \\ i_d[k] \\ i_q[k] \\ m_d[k-1] \\ m_q[k-1] \end{bmatrix} \in \mathbb{R}^8. \quad (8)$$

The terms $e_d[k]$ and $e_q[k]$ represent the direct and quadrature tracking errors. The previous errors provide a one-step temporal context. They approximate the discrete error derivative as follows:

$$\dot{e}_d[k] \approx \frac{e_d[k] - e_d[k-1]}{T_s}, \quad \dot{e}_q[k] \approx \frac{e_q[k] - e_q[k-1]}{T_s}. \quad (9)$$

The measured currents $i_d[k]$ and $i_q[k]$ capture the operating point of the plant. The previous modulation indices $m_d[k-1]$ and $m_q[k-1]$ encode the most recent feedback control action applied to the converter and provide an external recurrence path. In this case, the neural network controller is trained to predict the *incremental* modulation update $y[k] \in \mathbb{R}^2$:

$$y[k] = \begin{bmatrix} \Delta m_d[k] \\ \Delta m_q[k] \end{bmatrix} = \begin{bmatrix} m_d[k] - m_d[k-1] \\ m_q[k] - m_q[k-1] \end{bmatrix}. \quad (10)$$

The absolute control actions of the proposed neural network controller, applied to the converter's modulation stage, are recovered by accumulation:

$$m_d[k] = \text{sat}_{m_{\max}}(m_d[k-1] + \Delta m_d[k]). \quad (11)$$

$$m_q[k] = \text{sat}_{m_{\max}}(m_q[k-1] + \Delta m_q[k]). \quad (12)$$

where $\text{sat}_{m_{\max}}(\cdot)$ applies component-wise saturation to the admissible modulation interval $[-m_{\max}, m_{\max}]$. Note that the proposed neural network controller predicts incremental differences $(\Delta m_d, \Delta m_q)$ rather than absolute modulation indices. This external recurrence provides controller memory without a machine-learning model with recurrent connections; hence, the proposed model has fewer trainable parameters than recurrent neural networks such as LSTM or GRU. Different from a conventional PI implementation, the proposed controller does not contain a separate integral state whose output can continue to grow while the actuator is saturated, where additional anti-windup logic such as conditional integration or back-calculation is typically required [26]. Instead, the predicted modulation increment is accumulated directly in the modulation-index domain, and the accumulated command is clipped to the admissible modulation range. Therefore, Equations (11) and (12) should be interpreted as a modulation-index limiter with bounded external recurrence, rather than as a complete substitute for established anti-windup analysis under grid-fault or long-duration saturation conditions.

3.1. Practical Boundedness of the External Recurrence

The nonlinear closed-loop system formed by the MLP, the external accumulation, and the saturation operator does not admit the same global linear stability interpretation as a classical PI loop. Nevertheless, a practical boundedness property can be established for the nominal balanced-grid model used in this paper. Since the output activation is $\tanh(\cdot)$, each normalized network output is bounded. Let $\Delta m_{\max}^{\text{NN}}$ denote the maximum denormalized modulation increment allowed by the output scaling. Then,

$$|\Delta m_d[k]| \leq \Delta m_{\max}^{\text{NN}}, \quad |\Delta m_q[k]| \leq \Delta m_{\max}^{\text{NN}}. \quad (13)$$

The saturation in Equations (11) and (12) further guarantees

$$|m_d[k]| \leq m_{\max}, \quad |m_q[k]| \leq m_{\max}, \quad \forall k. \quad (14)$$

Thus, prediction errors cannot cause an unbounded modulation command through recursive accumulation. For the nominal dq -axis plant of Equation (2), the homogeneous current dynamics have eigenvalues $-R_f/L_f \pm j\omega$, whose real part is negative for $R_f > 0$. With bounded modulation commands, bounded grid voltage, and bounded reference signals inside the training envelope, the forced current response avoids mathematical divergence due to the algebraic limits imposed on the modulation vector. This argument establishes practical boundedness for the nominal balanced system evaluated here. It does not replace HIL, prototype, or weak-grid stability testing, and the corresponding boundaries are stated explicitly in Section 4.1.

3.2. Machine Learning Model

The proposed neural network controller was implemented using a multilayer perceptron (MLP) architecture of feedforward layers with dimensions 8–16–2. The hidden layer contains 16 neurons with hyperbolic tangent activations. The output layer contains two neurons with the same activation function. This design decision is motivated by the need to implement our proposed approach in resource-constrained environments. The neural network controller output is obtained using Equation (15) as follows:

$$\mathbf{y}[k] = \tanh(\mathbf{W}\mathbf{h} + \mathbf{b}^{(2)}) \in \mathbb{R}^2, \quad (15)$$

$$\mathbf{h} = \tanh(\mathbf{V}\mathbf{x}[k] + \mathbf{b}^{(1)}) \in \mathbb{R}^{16}, \quad (16)$$

where $\mathbf{x}[k] \in \mathbb{R}^8$ denotes the input, $\mathbf{V} \in \mathbb{R}^{16 \times 8}$ and $\mathbf{b}^{(1)} \in \mathbb{R}^{16}$ the hidden-layer weights and biases, and $\mathbf{W} \in \mathbb{R}^{2 \times 16}$ and $\mathbf{b}^{(2)} \in \mathbb{R}^2$ the output-layer weights and biases. The function $\tanh(\cdot)$ is applied element-wise as shown in Figure 3.

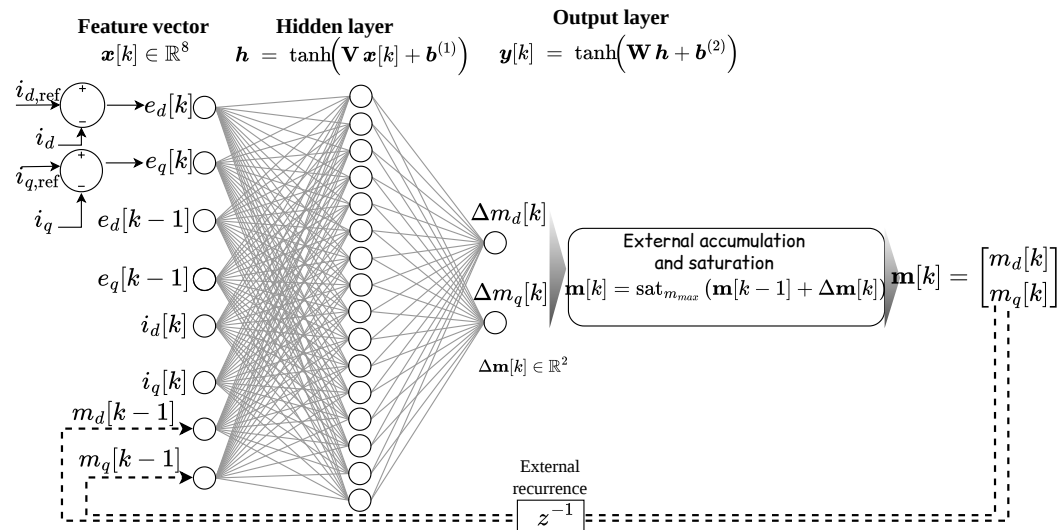


Figure 3. Proposed neural network controller with external recurrence. The dashed return path contains a one-sample delay, z^{-1} , supplying the previous modulation command $\mathbf{m}[k-1]$ to the dynamic update law.

The network has 178 trainable parameters in total, computed as

$$\underbrace{8 \times 16 + 16}_{\mathbf{V}, \mathbf{b}^{(1)}} + \underbrace{16 \times 2 + 2}_{\mathbf{W}, \mathbf{b}^{(2)}} = 178. \quad (17)$$

With single-precision floating-point storage, the complete trainable parameter set requires $178 \times 4 = 712$ bytes. The PSIM validation code uses double-precision constants for numerical convenience; in that form, the trainable parameters occupy $178 \times 8 = 1424$ bytes

before compiler-specific alignment. Table 2 summarizes the deterministic forward-pass cost of the deployed controller. The sampling period associated with $f_s = 70$ kHz is $T_s = 14.29 \mu\text{s}$. The values in Table 2 are theoretical arithmetic and storage counts for the stated architecture, not measured DSP cycle counts or execution latencies. The 160 multiply–accumulate operations describe the two matrix–vector products; feature normalization, output denormalization, external accumulation, and saturation require additional operations. A target-specific benchmark is still required before claiming real-time operation on a particular microcontroller or DSP.

Table 2. Theoretical arithmetic and storage requirements of the neural controller.

Operation or Storage Item	Count	Comment
Trainable parameters	178	$8 \times 16 + 16 + 16 \times 2 + 2$
Single-precision parameter memory	712 bytes	Excluding code, stack, and alignment
Input normalization constants	64 bytes	Eight means and eight inverse standard deviations in single precision
Matrix–vector MAC operations	160	128 hidden-layer MACs and 32 output-layer MACs
Bias additions	18	16 hidden-layer and 2 output-layer biases
Nonlinear activations	18	16 hidden-layer tanh and 2 output-layer tanh evaluations
Controller-state memory	8 scalar values	Current errors, measured currents, and previous modulation commands

MAC: multiply–accumulate. Counts do not specify execution time; normalization storage shown here covers the eight input features only.

For fixed-point implementation, the word lengths and scaling of the weights, activations, and stored modulation commands must preserve both dynamic range and the resolution of small modulation increments. Intermediate accumulators require sufficient headroom to avoid overflow, and approximating tanh introduces an additional numerical error. Quantization in the external recurrence can suppress small updates or produce limit cycles; therefore, closed-loop tracking accuracy and execution latency must be validated on the target processor. These effects were not assessed in the floating-point simulations reported here.

4. Materials and Methods

A typical VSC design [28] is used to demonstrate the performance of the proposed control strategy. The topology of a three-phase, six-switch active bridge is used, with a two-level modulation scheme via SVM PWM. The parameters and methods are briefly summarized in Table 3.

The PI controller used to generate the supervised targets was tuned to produce an underdamped fixed-gain current-loop response for the plant in Table 3. Under the evaluated reference-step trajectories, this design provides large initial control action but also exhibits under-damped excursions, overshoot, and abrupt modulation updates. The results reported in this paper therefore compare the proposed NNC with the same under-damped PI teacher under identical converter, grid, and reference-step conditions.

Table 3. Parameters of the three-phase grid-connected VSC simulation.

Parameter	Symbol	Value
Grid		
Line-to-line voltage (RMS)	$V_{g,LL}$	400 V
Grid frequency	f_0	60 Hz
DC Bus		
DC-link voltage	V_{dc}	800 V
Output Filter		
Filter inductance (per phase)	L_f	200 μ H
Inductor parasitic resistance	R_L	36 m Ω
Load		
Rated power	P_o	11 kW
Equivalent load resistance	R_o	58.2 Ω
SiC MOSFET Switches		
On-state drain-source resistance	$R_{ds(on)}$	30 m Ω
Body diode forward voltage	V_{fw}	3 V
Body diode on-state resistance	R_D	10 m Ω
Modulation and Switching		
Switching frequency	f_s	70 kHz
Modulation scheme	—	SVPWM
Dead time	t_{dead}	100 ns
PI Current Controller (d-q axis)		
Proportional gain	K_p	0.005
Integral time constant	T_i	0.25 ms
Tuning method	—	Blasko & Kaura [29]
Reference Limits		
d-axis current reference	$i_{d,ref}$	30 A
q-axis current reference	$i_{q,ref}$	−30 A
Simulation Settings		
Simulation time step	Δt	0.714 μ s
Solver	—	PSIM fixed-step solver

4.1. Operating Envelope and Scope of Validation

The training and evaluation data are restricted to the nominal balanced-grid VSC configuration summarized in Table 3. The reference commands vary across the generated trajectories, but the converter parameters, DC-link voltage, grid frequency, grid impedance, and ideal balanced-grid assumption remain fixed. Table 4 states the validation scope explicitly.

Table 4. Validation scope and unassessed practical conditions.

Aspect	Assessed in This Paper	Not Assessed in This Paper
Plant parameters	Nominal $L_f = 200 \mu\text{H}$, $R_L = 36 \text{ m}\Omega$, $V_{\text{dc}} = 800 \text{ V}$, $f_0 = 60 \text{ Hz}$	Filter drift, grid-frequency deviation, DC-link excursions, component aging, and grid-impedance sweeps
Grid condition	Balanced sinusoidal three-phase grid	Unbalance, negative-sequence components, harmonic distortion, voltage sags or swells, and PLL jitter
Controller task	Inner dq -axis current-loop reference tracking	Cascaded outer DC-link-voltage, active-power, or reactive-power loops
Implementation	Floating-point simulation and operation-count analysis	DSP/HIL timing, fixed-point quantization, ADC quantization, sensor noise, and prototype measurements
Saturation behavior	Component-wise modulation limiting in the nominal reference-step tests	Long-duration saturation under faults and comparison with back-calculation or conditional-integration anti-windup

4.2. Data Generation

In this work, the baseline control scheme in Figure 2 was implemented in PSIM [30] to control the three-phase VSC system described in Table 3. The simulations were performed with a fixed time step of $\Delta t \approx 0.714 \mu\text{s}$. A raw time-domain dataset

$$\mathcal{D}_{\text{raw}} = \left\{ \mathcal{T}^{(j)} \right\}_{j=1}^{N_{\text{sim}}}, \quad N_{\text{sim}} = 10,000,$$

was generated from independent PSIM simulations with randomly selected d - q axis current-reference changes. Each trajectory $\mathcal{T}^{(j)}$ contains the recorded time-domain signals

$$\mathcal{T}^{(j)} = \left\{ i_d^{(j)}[n], i_q^{(j)}[n], i_d^{\text{ref},(j)}[n], i_q^{\text{ref},(j)}[n], m_d^{(j)}[n], m_q^{(j)}[n] \right\}_{n=0}^{K_j}.$$

where i_d and i_q are the measured dq -axis currents, i_d^{ref} and i_q^{ref} are their corresponding reference trajectories, and m_d and m_q are the dq -axis modulation-index commands recorded from the baseline controller. Since the retained trajectories are generated by reference-command changes, the dataset primarily characterizes reference-tracking transients. It does not provide a complete disturbance-rejection dataset for grid sags, harmonics, DC-link perturbations, or load transients.

4.3. Proposed Event-Triggered Boundary-Detection Procedure

The proposed neural network controller was trained using supervised learning. The supervised-learning sample set was extracted from $\mathcal{D}_{\text{raw}}$ using the proposed event-triggered boundary-detection procedure. This procedure produced

$$\mathcal{S} = \left\{ \left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^{N_s}, \quad N_s = 11,000,000,$$

where $\mathbf{x}^{(i)}$ is the feature vector, and $\mathbf{y}^{(i)}$ is the corresponding target modulation increment for the i -th retained control-update boundary. Samples were retained only at the instants at which the digital PI controller updated its internal state. These instants are referred to as control-update boundaries. This selection reduces redundancy in the raw waveform data and aligns the supervised-learning samples with the actual decision instants of the digital controller.

To detect these boundaries, the integral-error states of both current axes were monitored as auxiliary continuous-time variables:

$$\begin{aligned} e_d^{\text{int}}(t) &= \int_0^t (i_d^{\text{ref}}(\tau) - i_d(\tau)) d\tau, \\ e_q^{\text{int}}(t) &= \int_0^t (i_q^{\text{ref}}(\tau) - i_q(\tau)) d\tau. \end{aligned} \quad (18)$$

In a digital implementation, the controller states and outputs exhibit discrete variations at each sampling instant. These variations were used to identify the update boundaries. Let

$$\mathbf{z}[k] = [m_d[k], m_q[k], e_d^{\text{int}}[k], e_q^{\text{int}}[k]]^\top$$

denote the monitored state vector at sample k . A control-update boundary is declared when

$$\max_j |\mathbf{z}_j[k] - \mathbf{z}_j[k-1]| > \varepsilon, \quad (19)$$

where $\varepsilon = 10^{-7} \|\mathbf{z}\|_\infty$ is a relative tolerance. The integral-error states were used only for boundary detection. They were not included in the neural network input vector $\mathbf{x}[k]$.

For each detected control-update boundary k_i , one supervised sample was formed by pairing the pre-update feature vector with the modulation increment produced at that update:

$$(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) = (\mathbf{x}[k_i], \mathbf{y}[k_i]),$$

where $\mathbf{x}[k]$ and $\mathbf{y}[k]$ are defined in Equations (8) and (10), respectively. Thus, the event-triggered procedure specifies the sampling instants, while the input–target construction follows the controller formulation of Section 3. As a result, explicit integral-error terms shown in Equation (18) are not required as network inputs. This point is a key distinction with respect to the previous neural-controller formulation [15]. In that formulation, integral-error quantities were treated as part of the learning input in order to provide temporal information to the network. In contrast, in our proposed controller, the integral-error states of Equation (18) are used only as auxiliary offline variables for detecting the digital control-update boundaries. They are not part of the deployed controller input vector.

All input features $\mathbf{x}[k]$ and target variables $\mathbf{y}[k]$ were standardized for numerical stability using the mean and standard deviation computed from the training set $\mathcal{D}_{\text{train}}$ only. The same statistics were then applied to the validation and test partitions.

4.4. Network Architecture and Training

Training was performed with the Adam optimizer at a learning rate of 10^{-3} , using a mini-batch size of 4096. The mean-squared-error loss was used to evaluate the learning process. The available data $\mathcal{D}$ were split using the holdout technique at the file level to avoid leakage among temporally correlated samples from the same simulation: 70% of the 10,000 simulation files were used for training and the remaining 30% for the test set $\mathcal{D}_{\text{test}}$. To adjust the hyperparameters, 15% of the training set $\mathcal{D}_{\text{train}}$ was used as the validation set $\mathcal{D}_{\text{val}}$. Early stopping with a patience of 200 epochs and a maximum of 2000 epochs was used to avoid overfitting. In practice, early stopping was triggered at approximately 600 epochs. The parameter set associated with the minimum validation loss was retained for the final evaluation.

4.5. Training Infrastructure

All training was performed on the High-Performance Computing infrastructure of Universidad San Francisco de Quito, Ecuador, using an NVIDIA DGX H200 system with

8 H200 Tensor Core GPUs. Inference validation was additionally performed on an NVIDIA DGX Spark (GB10 Grace Blackwell Superchip, 128 GB LPDDR5x). These resources were used for dataset processing, training, and hyperparameter exploration. They are not representative of the intended embedded deployment platform; therefore, the embedded feasibility claim in this paper is limited to the memory and operation counts reported in Table 2.

4.6. Performance Metrics

To provide a quantitative and reproducible basis for comparing the proposed neural network current controller (NNC) against the baseline proportional–integral (PI) controller, four standard transient-response evaluation metrics were employed. These metrics are widely adopted in the power-electronics and control-systems literature [31]. They are defined as follows:

- Percent overshoot, M_p quantifies the maximum excursion of the output beyond its steady-state value y_{ss} and reflects the damping properties of the controller.
- Settling time, t_s measures how rapidly the output enters and remains within a prescribed tolerance band around y_{ss} (the steady-state value of the output), and is therefore a reference-tracking metric for the evaluated step events.
- Integral of absolute error (IAE) accumulates the magnitude of the tracking error and offers a balanced measure that is less sensitive to isolated peaks.
- Steady-state error, e_{ss} captures the residual bias between the reference and the output once the transient has decayed, revealing any uncompensated DC offset introduced by the controller or by the modulation stage.

All four metrics, including IAE, are computed over event-synchronized time windows $t \in [t_{ev}, t_{ev} + T_w]$, where t_{ev} is the reference-step instant and T_w is the analysis horizon selected to include the transient response. In particular, IAE integrates $|r(t) - y(t)|$ over this event window, rather than over the entire simulation trajectory. For each paired reference-step event, the PI baseline and NNC use identical window endpoints. The same evaluation procedure is applied to the d and q axes.

The IAE integral is approximated by trapezoidal quadrature of the discretely sampled error $e[k] = r[k] - y[k]$. The steady-state output y_{ss} and reference r_{ss} are the arithmetic means over the final 10% of the window. The reference-step events used for transient evaluation are distinct from the digital control-update boundaries in Equation (19), which determine the supervised-learning sampling instants. Table 5 summarizes the metric definitions.

Table 5. Transient-response performance metrics.

#	Metric	Definition
1	M_p (%)	$\max\left(0, \frac{y_{\text{peak}} - y_{ss}}{ y_{ss} }\right) \times 100$
2	t_s (s)	Time to enter the $\pm 2\%$ band of y_{ss}
3	IAE	$\int_{t_{ev}}^{t_{ev} + T_w} r(t) - y(t) dt$
4	e_{ss}	$r_{ss} - y_{ss}$

5. Numerical Results and Discussion

This section presents a quantitative comparison between the proposed neural network current controller (NNC) and the baseline proportional–integral (PI) controller described in Section 2. The comparison relies on the per-step transient-response metrics defined

in Table 5, computed over the event-synchronized reference-step windows described in Section 4.6.

5.1. Regression Accuracy

Following the training protocol defined in Section 3, this subsection examines the regression quality of the 178 trainable parameters before the learned model is inserted into the VSC current-control loop. Figure 4 reports the learning curves, i.e., the evolution of the normalized mean-squared-error (MSE) loss as a function of training epoch. When both curves decrease together and stabilize at comparable values with a small persistent gap, the model generalizes adequately to unseen data. This indicates that the proposed model neither underfits nor overfits in the open-loop regression task of predicting the incremental modulation updates $(\Delta m_d, \Delta m_q)$ described in Equation (10). Both learning curves exhibit a rapid decay of approximately two orders of magnitude during the first ~ 30 epochs, followed by a slow monotonic decrease until the early-stopping criterion is triggered at epoch ≈ 600 . The minimum validation loss reached is 1.65×10^{-4} on the normalized MSE.

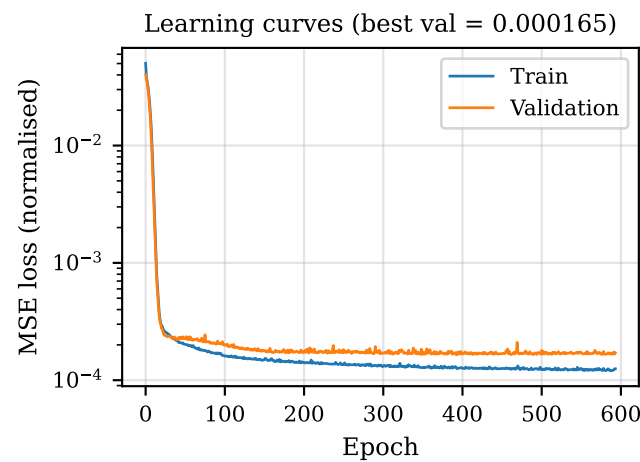


Figure 4. Learning curves of the proposed neural network controller during training. Logarithmic vertical scale. The best validation loss is 1.65×10^{-4} on the normalized MSE.

Once training has converged using the aforementioned early stopping at epoch ≈ 600 , a complementary evaluation was performed to verify that our proposed NNC with the obtained free parameters learned to map from inputs $x[k]$ (see Equation (8)) to outputs $y[k]$ (see Equation (10)). To assess how closely the NNC output $\hat{y}[k]$ matches the ground-truth reference $y[k] = [\Delta m_d[k], \Delta m_q[k]]^T$ on the test dataset $\mathcal{D}_{\text{test}}$, each $x[k]$ of $\mathcal{D}_{\text{test}}$ is presented to the trained NNC to obtain the output $\hat{y}[k]$, then using the ground truth $y[k]$ paired references, we evaluate the model as follows:

$$\mathbf{MAE}_{\hat{y}}(y[k], \hat{y}[k]) \triangleq \mathbb{E}[\|y[k] - \hat{y}[k]\|_1] \quad (20)$$

The prediction error on the held-out test set was quantified using the mean absolute error.

Figure 5 reports the parity plots of predicted versus reference modulation increments $(\Delta m_d$ and $\Delta m_q)$ on the held-out test set $\mathcal{D}_{\text{test}}$ (the 30% of simulation files never observed during training or validation). Both axes exhibit near-unity slope along the $y = x$ identity line, with mean absolute errors of 4.62×10^{-6} and 6.34×10^{-6} for the d - and q -axes respectively, and coefficients of determination $R_d^2 = 0.995$ and $R_q^2 = 0.997$. A mild deviation from linearity is visible at the extremities of the prediction range, attributable to the saturating nature of the tanh output activation. In closed loop, this deviation is constrained by the explicit modulation-interval saturation of Equations (11) and (12).

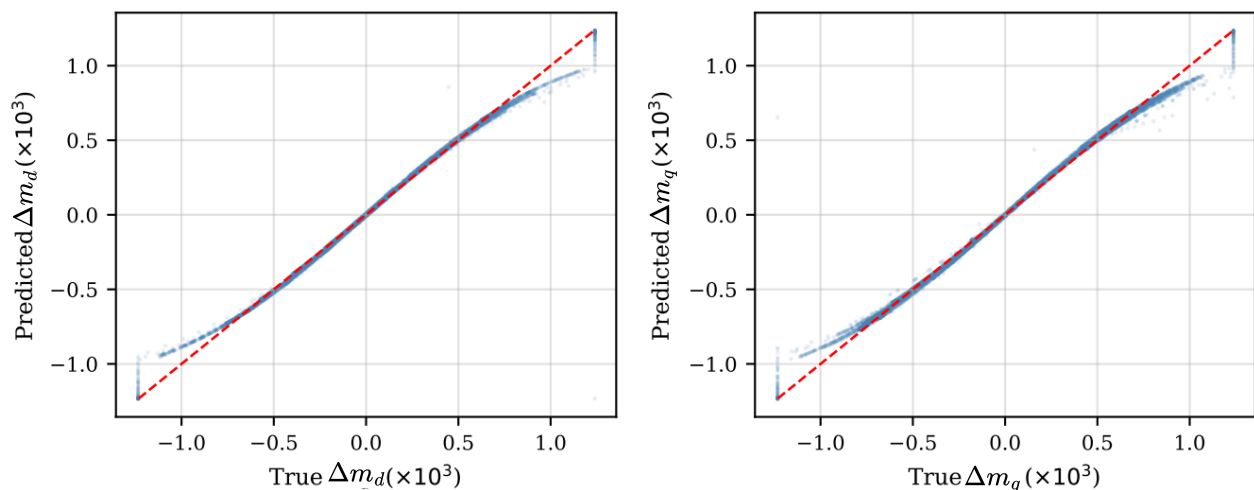


Figure 5. Regression accuracy of NNC. Parity plots on the held-out test set $\mathcal{D}_{\text{test}}$ for the predicted modulation increments. **(Left)** Δm_d ($\text{MAE} = 4.62 \times 10^{-6}$, $R^2 = 0.995$). **(Right)** Δm_q ($\text{MAE} = 6.34 \times 10^{-6}$, $R^2 = 0.997$). The dashed red line denotes the ideal identity $y = x$, and the blue dots represent the individual prediction samples from the test set.

5.2. Recovering a Better-Damped Current Loop from an Under-Damped PI Teacher

The supervised targets are generated from an under-damped PI teacher, so the NNC should not be interpreted as a controller trained by direct closed-loop performance optimization. Instead, the learning task can be understood as recovering the useful current-loop action of the teacher while filtering out part of the oscillatory and impulsive behavior visible in its modulation trace. First, the target is the complete modulation increment $\Delta m_{d,q}$ after feed-forward, decoupling, limiting, and digital update effects rather than the ideal PI increment alone. Second, the compact 8–16–2 architecture acts as a smooth nonlinear regression model; it cannot reproduce high-frequency or abrupt sample-to-sample variations of the PI trace exactly. Third, the tanh output and the external modulation-domain saturation bound the increment and the accumulated command. Finally, the previous modulation commands included in the input vector make the deployed controller a dynamic update law, even though the neural model itself is feed-forward.

For these reasons, the NNC can produce a smoother and better-damped trajectory than the original PI trace even when trained from PI-generated data. The comparison in this section therefore evaluates the closed-loop behavior of this learned, regularized, and bounded incremental policy under the same reference-step evaluation procedure used for the PI baseline. The improvement should be read as recovery of the desired current-loop behavior from an under-damped teacher, not as proof that the NNC is universally superior to all possible PI tunings. It also does not replace an ablation against alternative limiters, anti-windup strategies, or model-predictive controllers.

5.3. Closed-Loop Transient Comparison

This subsection evaluates the proposed NNC as the inner current controller in the VSC simulation workflow. Figure 6 compares the closed-loop response of the proposed NNC and the PI baseline for a representative step transition within the test set. The d -axis response provides the clearest transient improvement. The NNC reaches the $\pm 2\%$ settling band in 1.35 ms, whereas the PI teacher requires 49.47 ms for the same transition, reducing the overshoot from 8.53% to 3.20%. The q -axis transition highlights a complementary effect. The PI baseline enters the $\pm 2\%$ band earlier, with $t_s = 0.67$ ms compared with $t_s = 1.63$ ms for the NNC. However, this earlier band crossing is obtained with a larger transient excursion. The PI overshoot reaches 11.26% and is followed by a pronounced

undershoot, whereas the NNC limits the overshoot to 7.18% and produces a smoother approach to the reference. This behavior is consistent with the proposed interpretation: the neural controller preserves the useful corrective tendency of the teacher while improving damping through the learned bounded incremental modulation law.

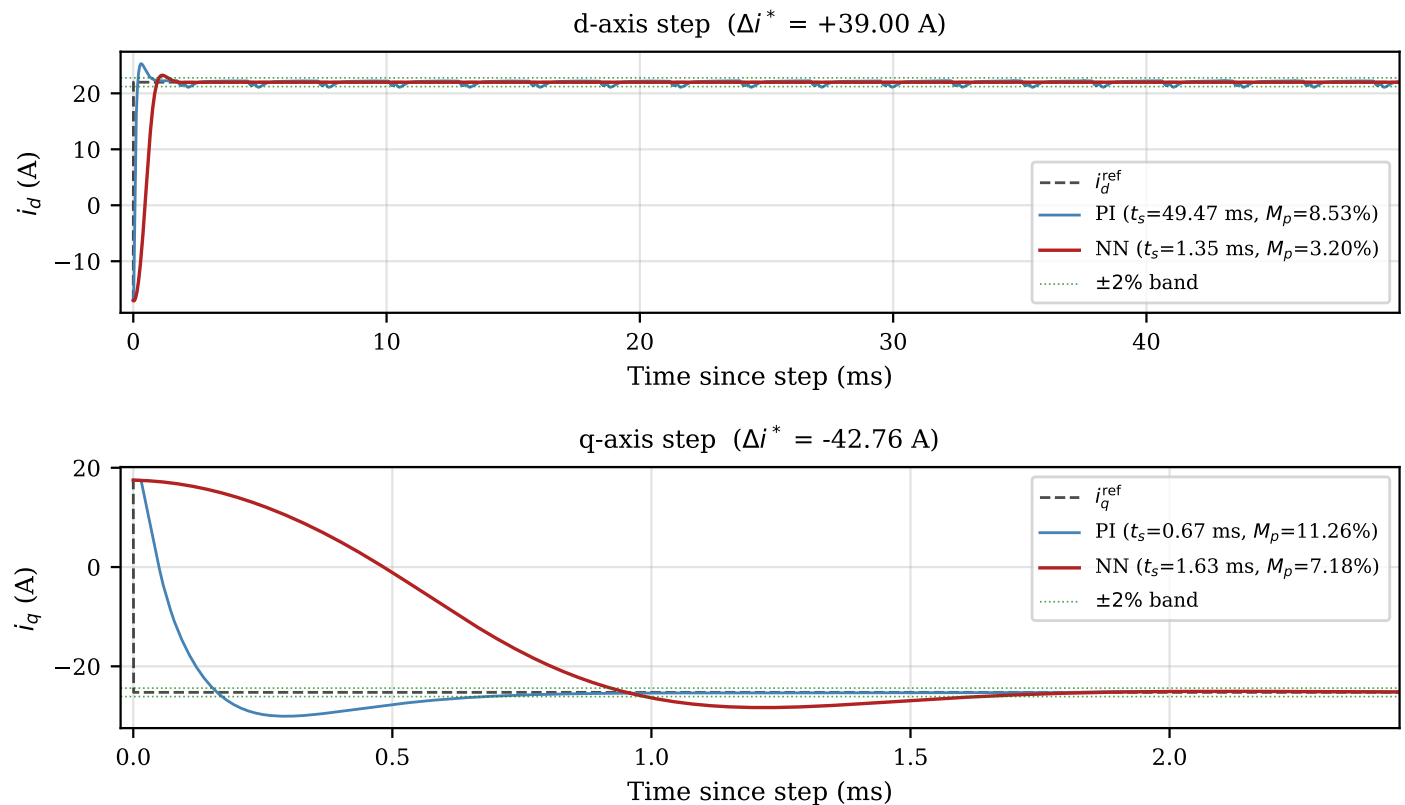


Figure 6. Closed-loop transient response of the PI baseline (blue) and the proposed NNC (red) for a representative step transition. **(Top)** d -axis reference step $\Delta i_d^{\text{ref}} = +39.00$ A. **(Bottom)** q -axis reference step $\Delta i_q^{\text{ref}} = -42.76$ A. Dashed lines indicate the current references, and green dotted lines denote the $\pm 2\%$ settling band. The annotations t_s and M_p denote settling time (ms) and percent overshoot (%), respectively.

To verify that the per-step improvement persists under multi-step reference excitation, Figure 7 contrasts the two controllers over a rigorous 500 ms multi-step trajectory, containing nine successive reference changes on each synchronous axis. Across the complete trajectory, the NNC tracks both i_d^{ref} and i_q^{ref} with negligible steady-state bias and avoids the large-amplitude transient spikes generated by the under-damped PI teacher at the reference-update instants. The bottom two panels of Figure 7 report the accumulated modulation commands at discrete time step k . The plotted variables $u_d[k]$ and $u_q[k]$ denote the accumulated modulation commands $m_d[k]$ and $m_q[k]$, respectively. The NNC with the proposed external recurrence reduces impulsive controller updates during large reference transitions at the step instants, from ± 0.22 produced by the PI baseline controller to ± 0.05 obtained with our proposed NNC. The reduction in abrupt modulation excursions is consistent with the bounded incremental-update formulation, which limits the maximum change in the modulation vector at each controller update.

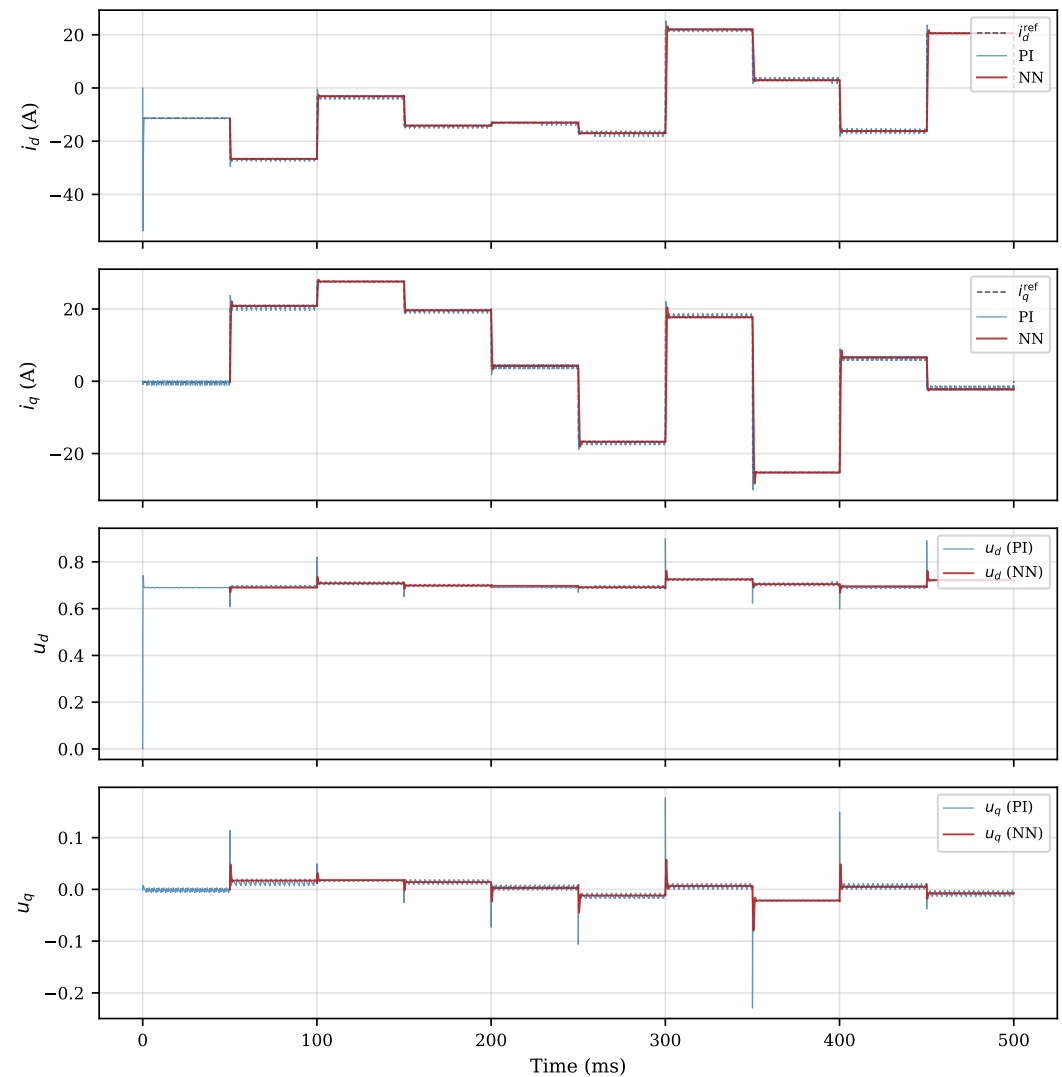


Figure 7. Full-trace comparison over a 500 ms multi-step trajectory. From top to bottom: d -axis current, q -axis current, d -axis modulation index, and q -axis modulation index. Blue traces correspond to the PI baseline, red traces to the proposed NNC, and dashed lines to the references. The plot labels u_d and u_q denote m_d and m_q , respectively.

5.4. Statistical Comparison

The statistical comparison was performed on the held-out test partition, corresponding to 30% of the 10,000 independent PSIM simulations used for data generation. Since the complete event-triggered dataset contains 11,000,000 samples, this partition contains approximately 3.3×10^6 test samples before transient-level aggregation. From these test trajectories, synchronized step-transition windows were extracted. Each window was evaluated with the same procedure for the PI baseline and for the proposed neural network controller (NNC), producing paired transient-response metrics for M_p , t_s , IAE, and $|e_{ss}|$. Table 6 reports the mean performance metrics for the held-out test evaluation and the median values displayed in Figure 8. The rightmost column retains the reported relative change Δ in the mean; the medians are transcribed at the precision shown in the figure.

The NNC reduces the averaged overshoot from 19.50% to 8.05%, corresponding to a 58.73% reduction. The averaged settling time decreases from 42.56 ms to 2.74 ms, corresponding to a 93.57% reduction. The averaged IAE decreases from 0.0132 to 0.0096 (27.45% reduction), and the averaged steady-state error magnitude decreases from 0.0238 to 0.0137 (42.31% reduction). The NNC achieves lower overshoot than the PI in 96.2%, a shorter settling time in 82.1%, a smaller integral absolute error in 74.5%, and a reduced steady-state

error magnitude in 66.0% of the cases. These statistics support the interpretation that the network learned to recover the useful current-loop behavior embedded in the teacher data while improving damping and reducing impulsive modulation updates. Figure 8 complements the average results by showing the distributions of the four reported metrics for the 106 plotted step events.

Table 6. Mean and median transient metrics. Median values match the annotations in Figure 8; Δ is the reported relative change in the mean.

Metric	PI Baseline		NNC		Mean Change
	Mean	Median	Mean	Median	Δ (%)
M_p (%)	19.4995	11.32	8.0471	5.674	−58.73
t_s (ms)	42.5637	49.67	2.7361	1.351	−93.57
IAE	0.0132	0.01472	0.0096	0.006924	−27.45
$ e_{ss} $	0.0238	0.02207	0.0137	0.01297	−42.31

Medians are reproduced at the precision displayed for the 106 step events in Figure 8. Mean values and their reported relative changes are retained from the original analysis.

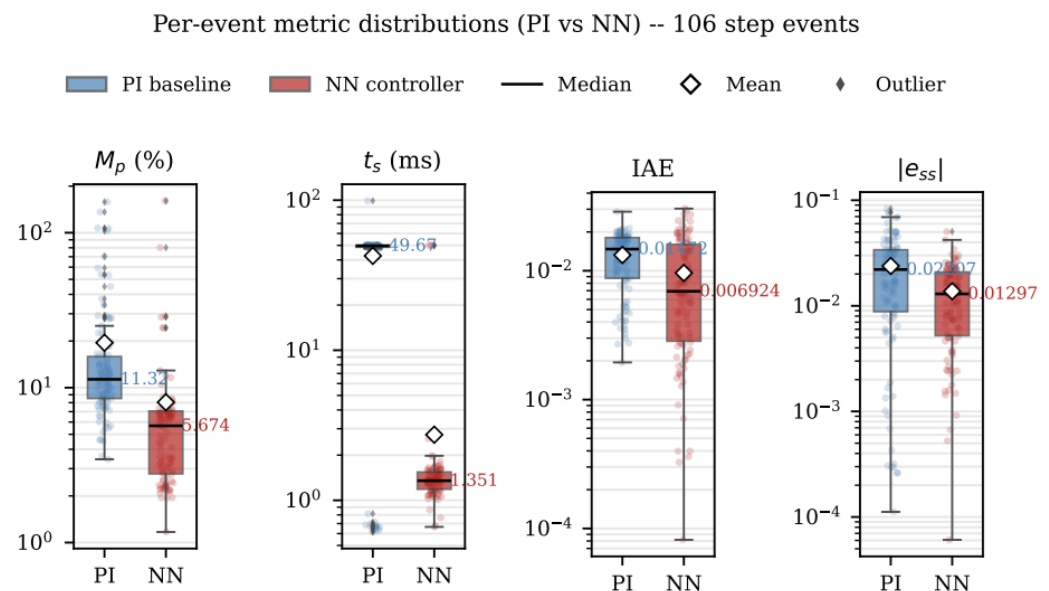


Figure 8. Per-event metric distributions for the PI baseline and proposed NNC (106 step events, as indicated in the plot). Black horizontal lines denote medians, with their numerical values annotated beside the boxes; diamonds denote means. The metric labels are percent overshoot M_p , settling time t_s , integral of absolute error (IAE), and steady-state error magnitude $|e_{ss}|$. The annotated medians are also listed in Table 6.

To verify that the reported improvements are not dominated by a single axis, Table 7 disaggregates the averaged metrics by synchronous axis. The improvements are consistent across both axes for overshoot, IAE, and steady-state error.

Table 7. Axis-wise averaged performance metrics.

Metric	d -Axis			q -Axis		
	PI	NN	Δ (%)	PI	NN	Δ (%)
M_p (%)	19.4873	7.1054	−63.54	19.5121	9.0249	−53.75
t_s (ms)	44.9026	1.2546	−97.21	40.1348	4.2746	−89.35
IAE	0.0146	0.0083	−43.09	0.0118	0.0110	−7.44
$ e_{ss} $	0.0272	0.0125	−53.89	0.0203	0.0150	−26.25

The settling-time improvement is substantial on both axes: -97.21% on the d-axis and -89.35% on the q-axis. The overshoot, IAE, and steady-state error reductions also remain consistent across both synchronous axes.

5.5. Limitations and Practical Interpretation

The results in this section quantify reference-step tracking for the nominal balanced-grid configuration defined in Table 3. They do not quantify disturbance rejection against grid-voltage sags or swells, injected harmonics, DC-link perturbations, load transients, measurement noise, ADC quantization, or PLL jitter. They also do not include parameter-sweep robustness tests for L_f , R_f , DC-link voltage, grid impedance, or grid frequency. Therefore, the reliability of the trained network without retraining should be considered established only inside the nominal operating envelope stated in Table 4.

The proposed modulation-domain saturation prevents recursive growth of the accumulated command, as discussed in Section 3.1. However, sustained saturation during grid faults, comparison with classical anti-windup designs, and interaction with outer DC-link-voltage or active/reactive-power loops remain open implementation issues. Similarly, Table 2 shows that the controller has a small deterministic operation count, but fixed-point quantization, execution latency, and memory layout must be benchmarked on the final DSP or microcontroller before industrial deployment.

6. Conclusions

This paper presented a compact neural network current controller for a three-phase, grid-connected voltage-source converter. The proposed controller replaces the inner dq -axis PI compensator while preserving the surrounding grid-following architecture, including grid synchronization, reference-frame transformations, and the modulation stage. The controller is implemented as a single-hidden-layer feed-forward multilayer perceptron that predicts incremental dq -axis modulation updates, which are then accumulated and saturated in the modulation-index domain.

The main contribution of this work is to show that simulation-based closed-loop dynamic current regulation can be achieved without internally recurrent neural network layers. Instead, the required temporal information is supplied through causal controller variables. Therefore, the proposed controller should be interpreted not as a purely memory-less static map but as a compact feed-forward network embedded in a dynamic update law. This structure requires only 178 trainable parameters, corresponding to 712 bytes in single-precision storage, and its deterministic forward pass requires 160 multiply–accumulate operations plus 18 tanh evaluations.

The proposed controller was trained from 11×10^6 event-synchronized samples extracted from 10,000 independent PSIM simulations and evaluated on held-out trajectories. Compared with the under-damped PI teacher, the neural network controller reduced the average overshoot by 58.73%, settling time by 93.57%, IAE by 27.45%, and steady-state error magnitude by 42.31% in the evaluated reference-step tests. These performance gains apply only to the nominal balanced-grid simulation envelope defined in Table 3 and Section 4.1. They do not establish performance under unbalanced or disturbed grids, parameter variations, or embedded hardware operation. These reductions indicate that the learned bounded incremental policy can retain useful corrective behavior present in the teacher data while suppressing part of its under-damped transient response.

The present study remains a simulation-only proof of concept. Future work should benchmark the bounded external recurrence against conventional anti-windup schemes, including back-calculation and conditional integration, under identical modulation limits and saturation scenarios. Further work will focus on robustness under parameter drift and

weak-grid conditions, disturbance-rejection tests, fixed-point and latency benchmarking on a target DSP, hardware-in-the-loop or experimental validation, and quantifying how the improved inner current-loop response affects cascaded outer-loop regulation.

Author Contributions: Conceptualization, B.A.A.A., F.G. and J.C.R.; Methodology, B.A.A.A., F.G. and J.C.R.; Software, B.A.A.A.; Validation, B.A.A.A. and J.C.R.; Formal analysis, B.A.A.A. and F.G.; Investigation, B.A.A.A.; Resources, B.A.A.A.; Data curation, B.A.A.A. and J.C.R.; Writing—original draft, B.A.A.A. and J.C.R.; Writing—review & editing, B.A.A.A., F.G. and J.C.R.; Supervision, J.C.R.; Project administration, F.G. and J.C.R.; Funding acquisition, F.G. and J.C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Universidad San Francisco de Quito (USFQ) under Grant 39820, with funding provided by Analog Devices, Inc. (ADI). This work was also supported by the São Paulo Research Foundation, FAPESP (Grant 2025/01540-6 and thematic project 2021/11380-5).

Data Availability Statement: The simulation settings, controller equations, network architecture, training protocol, and performance-metric definitions are documented in this article. The accompanying manuscript materials do not include the underlying PSIM project files, sample- or event-level datasets, trained network weights, or normalization constants.

Acknowledgments: The authors gratefully acknowledge Analog Devices, Inc. for all the support in the development of this project. The authors acknowledge the Sigma AI Lab at Universidad San Francisco de Quito (USFQ) for providing the computational resources and AI infrastructure that enabled this research. The authors also acknowledge support of São Paulo Research Foundation, FAPESP (Grant 2025/01540-6 and thematic project 2021/11380-5).

Conflicts of Interest: Author Juan Carlos Rodríguez was employed by the company Analog Devices Inc. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Luo, J.; Xiao, D. Advances in renewable energy power converters: A review of multilevel topologies and design criteria. *Renew. Sustain. Energy Rev.* **2026**, *226*, 116261. [\[CrossRef\]](#)
2. Li, S.; Wang, H.; Huang, Y.; He, G.; Liu, C.; Wang, W. Dynamic Interaction and Stability Analysis of Grid-following Converter Integrated Into Weak Grid. *CSEE J. Power Energy Syst.* **2025**, *11*, 552–566. [\[CrossRef\]](#)
3. Chowdhury, V.R. Internal Model Based Grid Voltage Estimation and Control of a Three-Phase Grid Connected Inverter for PV Application. *IEEE Trans. Energy Convers.* **2021**, *36*, 3568–3577. [\[CrossRef\]](#)
4. Kabalci, E. *Multilevel Inverters: Control Methods and Advanced Power Electronic Applications*; Academic Press: Cambridge, MA, USA, 2021.
5. Yazdani, S.; Ferdowsi, M.; Shamsi, P. Internal model based smooth transition of a three-phase inverter between islanded and grid-connected modes. *IEEE Trans. Energy Convers.* **2019**, *35*, 405–415. [\[CrossRef\]](#)
6. Gao, X.; Zhou, D.; Anvari-Moghaddam, A.; Blaabjerg, F. Stability analysis of grid-following and grid-forming converters based on state-space modelling. *IEEE Trans. Ind. Appl.* **2024**, *60*, 4910–4920. [\[CrossRef\]](#)
7. Xu, C.; Zou, Z.; Yang, J.; Wang, Z.; Chen, W.; Buticchi, G. Transient stability analysis and enhancement of grid-forming and grid-following converters. *IEEE J. Emerg. Sel. Top. Ind. Electron.* **2024**, *5*, 1396–1408. [\[CrossRef\]](#)
8. Sun, L.; You, F. Machine learning and data-driven techniques for the control of smart power generation systems: An uncertainty handling perspective. *Engineering* **2021**, *7*, 1239–1247. [\[CrossRef\]](#)
9. Li, Y.; Gu, Y.; Green, T.C. Revisiting grid-forming and grid-following inverters: A duality theory. *IEEE Trans. Power Syst.* **2022**, *37*, 4541–4554. [\[CrossRef\]](#)
10. Ali, Z.; Ahmed, A.H.; Saleh, Z.H.; Raisz, D. Review of Control Strategies in Three Phase Grid Connected Renewable Energy Systems with Practical Case Study. *Results Eng.* **2026**, *29*, 109286. [\[CrossRef\]](#)
11. Adiche, S.; Larbi, M.; Toumi, D.; Bouddou, R.; Bajaj, M.; Bouchikhi, N.; Belabbes, A.; Zaitsev, I. Advanced control strategy for AC microgrids: A hybrid ANN-based adaptive PI controller with droop control and virtual impedance technique. *Sci. Rep.* **2024**, *14*, 31057. [\[CrossRef\]](#)
12. Zhao, S.; Blaabjerg, F.; Wang, H. An overview of artificial intelligence applications for power electronics. *IEEE Trans. Power Electron.* **2020**, *36*, 4633–4658. [\[CrossRef\]](#)

13. Li, S.; Fairbank, M.; Johnson, C.; Wunsch, D.C.; Alonso, E.; Proao, J.L. Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *25*, 738–750. [CrossRef]
14. Bakeer, A.; Mohamed, I.S.; Malidarreh, P.B.; Hattabi, I.; Liu, L. An artificial neural network-based model predictive control for three-phase flying capacitor multilevel inverter. *IEEE Access* **2022**, *10*, 70305–70316. [CrossRef]
15. Ghidewon-Abay, S.; Mehrizi-Sani, A. Control of a three-phase grid-connected voltage-sourced converter using long short-term memory networks. *Energies* **2022**, *16*, 453. [CrossRef]
16. Kazmierkowski, M.P.; Malesani, L. Current control techniques for three-phase voltage-source PWM converters: A survey. *IEEE Trans. Ind. Electron.* **2002**, *45*, 691–703. [CrossRef]
17. Bahrani, B.; Kenzelmann, S.; Rufer, A. Multivariable-PI-based dq current control of voltage source converters with superior axis decoupling capability. *IEEE Trans. Ind. Electron.* **2010**, *58*, 3016–3026. [CrossRef]
18. Parvez, M.; Elias, M.; Rahim, N.; Osman, N. Current control techniques for three-phase grid interconnection of renewable power generation systems: A review. *Sol. Energy* **2016**, *135*, 29–42. [CrossRef]
19. Hong, J.; Zhang, X.; Cao, R. Robust model predictive control for three-level voltage source inverters. *J. Power Electron.* **2021**, *21*, 747–756. [CrossRef]
20. Sun, J.; Qiu, L.; Liu, X.; Ma, J.; Rodríguez, J.; Fang, Y. Model-free moving-discretized-control-set predictive control for three-phase dual-active-bridge converters. *IEEE Trans. Power Electron.* **2023**, *39*, 9160–9173. [CrossRef]
21. Lai, N.B.; Kim, K.H. Robust control scheme for three-phase grid-connected inverters with LCL-filter under unbalanced and distorted grid conditions. *IEEE Trans. Energy Convers.* **2017**, *33*, 506–515. [CrossRef]
22. Kim, S.D.; Tran, T.V.; Yoon, S.J.; Kim, K.H. Current controller design of grid-connected inverter with incomplete observation considering L-/LC-type grid impedance. *Energies* **2024**, *17*, 1855. [CrossRef]
23. Stojić, D.M.; Šekara, T.B. Multi-resonant current controller for grid-tied inverters with LCL filters based on pole placement under robustness constraints. *J. Power Electron.* **2025**, *25*, 1942–1953. [CrossRef]
24. Blaabjerg, F.; Teodorescu, R.; Liserre, M.; Timbus, A.V. Overview of control and grid synchronization for distributed power generation systems. *IEEE Trans. Ind. Electron.* **2006**, *53*, 1398–1409. [CrossRef]
25. Holmes, D.G.; Lipo, T.A. *Pulse Width Modulation for Power Converters: Principles and Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2003.
26. Texas Instruments. *C2000 Digital Control Library User's Guide*; Version 3.4; Texas Instruments: Dallas, TX, USA, 2020.
27. Yepes, A.G.; Freijedo, F.D.; Doval-Gandoy, J.; López, Ó.; Malvar, J.; Fernandez-Comesana, P. Effects of discretization methods on the performance of resonant controllers. *IEEE Trans. Power Electron.* **2010**, *25*, 1692–1712. [CrossRef]
28. Rodríguez, J.C. An 11-kW bidirectional on-board charger with highly integrated SiC-cascode technology. In *Proceedings of the 2024 IEEE Power and Energy Conference at Illinois (PECI)*; IEEE: New York, NY, USA, 2024; pp. 1–5.
29. Blasko, V.; Kaura, V. A new mathematical model and control of a three-phase AC-DC voltage source converter. *IEEE Trans. Power Electron.* **2002**, *12*, 116–123.
30. Altair Engineering, Inc. Power Electronics and Motor Drive Software—Altair PSIM. 2026. Available online: <https://altair.com/p-sim> (accessed on 11 May 2026).
31. Nise, N.S. *Control Systems Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2019.

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.