Citation (APA)
Majstorović, M., Brković, B., Ristić, L., & Lekić, A. (2026). Neural Network Assisted Model Predictive Control of a Modular Multilevel Converter. *IEEE Open Access Journal of Power and Energy*, *13*, 619 - 627. https://doi.org/10.1109/OAJPE.2026.3718999

IEEE Open Access Journal of
Power and Energy

Received 24 September 2025; revised 28 March 2026; accepted 24 July 2026. Date of publication 3 August 2026; date of current version 7 August 2026.

Digital Object Identifier 10.1109/OAJPE.2026.3718999

Neural Network-Assisted Model Predictive Control of a Modular Multilevel Converter

MILOVAN MAJSTOROVIĆ [1], BOGDAN BRKOVIĆ [1], LEPOSAVA RISTIĆ [1] (Member, IEEE), AND ALEKSANDRA LEKIĆ [2] (Senior Member, IEEE)

[1]School of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia
[2]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, GA 2600, The Netherlands

CORRESPONDING AUTHOR: B. BRKOVIĆ (brkovic@etf.bg.ac.rs)

This work was supported by European Union's HORIZON-WIDERA-2021-ACCESS-03 under Grant 101079200.

ABSTRACT This paper describes an advanced algorithm for current control in a modular multilevel power converter. The proposed algorithm is based on deadbeat model predictive control, which is further enhanced by applying an artificial neural network. Rather than supplanting the existing predictive controller, the neural network is used to supplement it, thereby improving control performance. The reference current tracking capability is enhanced under all operating conditions, particularly in cases where the reactive elements of the converter exhibit a relatively high resistive component. By introducing the neural network, the inherent delay of model predictive control in such scenarios is virtually eliminated, which reduces the overall control error. The network is trained offline on a set of control inputs obtained by optimizing the current response in a wide range of operating modes. Algorithm verification is performed for a three-phase modular multilevel converter in different steady-state and transient conditions. A state-of-the-art high-fidelity real-time simulator is used in both the algorithm development and verification stages.

INDEX TERMS Current control, model predictive control, modular multilevel converter, neural network, real-time simulator.

I. INTRODUCTION

The modular multilevel converter (MMC) is a promising technology for high- and medium-voltage, high-power direct current applications, such as high-voltage DC (HVDC) transmission, renewable energy generation, electric propulsion, and storage systems [1]. Despite its numerous advantages over competitive topologies, including better controllability, unity power factor operation, and lower current harmonic distortion, the MMC remains a challenging topology in terms of design and control complexity. One of the main aspects to consider is AC current control, which depends strongly on the circuit parameters, particularly the grid-side and MMC arm impedances. Namely, depending on the resistive component of the impedance, conventional AC current controllers cannot successfully track the reference value. Even more advanced control techniques, such as MPC [2], [3], sliding mode control [3], backstepping control [3], and others, are often not sufficiently precise when the grid-side or arm impedances are variable or comprise a significant resistive component.

Due to their enhanced adaptability and robustness, machine learning algorithms [4], [5] are expected to improve control performance in such operating conditions. MPC is a powerful control for switching devices, such as MMCs [6], [7], [8], driven by switching pulses generated by a cost function solution. The applicability of MPC has been demonstrated across various applications, such as hierarchical grid-forming control [7]. Similarly, MPC shows great promise for grid-following control applications [8]. It is important to note that these works [7], [8] only support outer control loops. The cost function describes the optimization problem with multiple optimization variables, processed within microseconds, which proves suitable even for lower-level control loops, such as capacitor voltage balancing [2]. The application of machine learning in combination with MPC therefore further improves the already good performance of MPC and potentially reduces the computational burden. In the context of integrating neural networks with model predictive control (MPC), the literature can be broadly categorized into three groups:

TABLE 1. Summary of NN-assisted / NN-emulated predictive control for power converters.

Author, year [ref]	Key takeaway	Topology	NN role in control algorithm
D. Wang, 2021 [5]	ANN–MPC on FCMLI cuts FPGA resources while keeping MPC-level performance	3-phase FC multi-level inverter	NN *emulator* of virtual MPC (ANN replaces optimizer in real-time controller)
Wang, 2021 [6]	NN regression vs. pattern recognition to emulate MPC for MMC; regression gives best performance and lowest burden	MMC (voltage-source)	NN *emulator* of FCS–MPC (directly outputs insertion indices; MPC only used offline for training)
Liu, 2020 [9]	Predictor-based NN + FCS–MPC for MMC removes need for model parameters and weighting factors	MMC	NN *predictor/model* of converter dynamics; FCS–MPC still evaluates candidates
Liu, 2022 [10]	Data-driven NN predictors + Lyapunov-based FCS–MPC; robust AFE–MMC control without explicit model/weights	Active front-end MMC	NN *predictor/model* providing multi-step state forecasts inside FCS–MPC (MPC law preserved)
Wu, 2025 [11]	Dynamic-linearization model with parameters updated by Adaline NN supplies predictions for FCS–MPC	3-level NPC inverter	Adaline NN *online identifier/predictor* of model parameters for FCS–MPC (assists MPC, not replacing it)
Liu, 2023 [12]	Robust data-driven MFAC+FCS–MPC framework; ANN imitator replaces it to cut online complexity (MMC case)	MMC (general power converter framework)	(i) NN *predictor* in MFAC+FCS–MPC; (ii) separate ANN *emulator* of this robust predictive controller
Sabzevari, 2021 [13]	State-space RNN with PSO used as model-free predictive controller, more robust than model-based FCS–MPC	3-phase grid-connected converter	NN *model-free predictive controller* (NN predicts future current inside cost minimization)
S. Wang, 2021 [14]	ML-based MPC for MMC; ANN trained from fast MPC achieves same performance with much lower online cost	MMC	NN *emulator* of fast MPC (direct policy approximation)
Mohamed, 2019 [15]	ANN trained by MPC expert; low-THD LC-inverter control without MPC online	2-level 3-phase VSI with LC filter	NN *emulator* of MPC (feedforward ANN generates optimal switching vector)
Singh, 2023 [16]	Multi-layer NN approximates MMC–MPC behavior, reducing real-time computational effort	MMC	NN *emulator* of MPC cost-minimizing action (regression on MPC dataset)
Yousefi, 2023 [17]	Direct and indirect MPC for MMC compared; ANN offers reduced burden with similar behavior	MMC	NN *emulator* of DMPC/IMPC (trained from both predictive schemes)
Malidarreh, 2021 [18]	ANN-based, model-free strategy trained by MPC improves robustness and keeps low THD on FCMLI	3-phase 4-level FC inverter	NN *emulator* of MPC (data-driven controller mapping states to optimal switching)
Novak, 2021 [19]	Supervised imitation learning of FS–MPC; ANN imitator keeps performance with significantly lower burden	Stand-alone converter (FS–MPC benchmark)	NN *emulator* of FS–MPC (policy network trained on FS–MPC labels)
H. Wang, 2023 [20]	BP-NN trained on MPC data controls MMC; similar performance, lower computational cost	MMC	NN *emulator* of MPC (trained offline based on MPC data; NN used to generate control actions)
Simonetti, 2023 [21]	NN-based FCS–MPC for CHB STATCOM overcomes computational growth with many levels; FPGA-validated	Cascaded H-bridge STATCOM	NN *emulator* of FS–MPC (ANN approximates optimal switching decision)
Gong, 2024 [22]	ANN-implemented MPC for MMC rectifier improves grid adaptability under non-ideal grids	MMC-based active rectifier	NN *emulator/implementation* of MPC law (data-driven replacement of model-dependent optimizer)

i) Neural networks acting as data-driven predictors within FCS-MPC, while the control law remains MPC-based [9], [10], [11],
ii) Neural networks employed for model or parameter adaptation in conjunction with MPC [12], [13], and
iii) Neural networks fully replacing the MPC controller [5], [6], [14], [15], [16], [17], [18], [19].

More information on research related to neural network–assisted or emulated MPC is provided in Table 1.

This paper presents an MMC current control algorithm based on MPC, with a corrective action implemented via an artificial neural network (NN). The application of NN further improves the operation of the deadbeat MPC presented in [2]. Contrary to the deadbeat control research carried out in [23], [24], and [25], the control system proposed in [2] considers non-linearities in the AC current prediction model. However, this control mechanism cannot accurately track the reference current in all plausible scenarios. Namely, the deviation between the obtained and reference current increases when the resistive component of the filter becomes non-negligible. The introduction of the NN significantly reduces this error, with no considerable increase in computational burden. It should be noted that the approach proposed in this paper, unlike those presented in [15] and [25], does not replicate the MPC algorithm using a neural network,

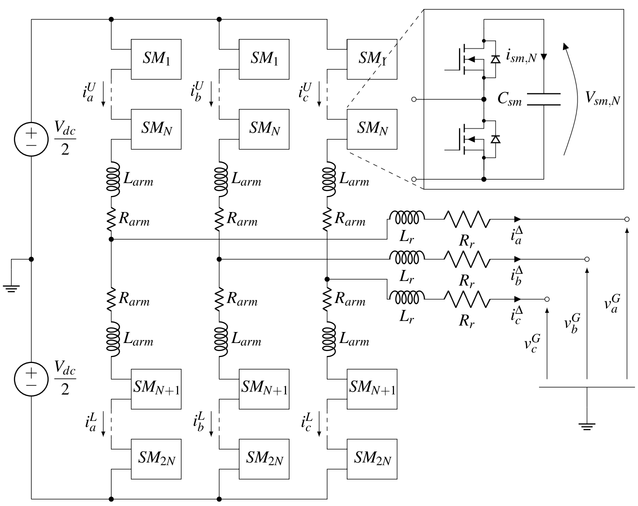


FIGURE 1. Three-phase MMC structure.

but rather supplements it to improve overall performance. Therefore, the proposed controller, with MPC at its core, still relies on the physical model, while the neural network primarily serves as a corrective action. In this way, the MPC's inherent feedback feature is retained, and an NN with a relatively simple structure is sufficient. The effectiveness of the proposed approach is demonstrated through real-time simulation on a state-of-the-art RTDS device, showing that the algorithm operates properly even under time-constrained conditions at a sampling frequency of 1 kHz.

The paper is organized as follows. Section II describes the MPC algorithm. The addition of NN is described in Section III, including details on the control optimization and implementation of NN-MPC. Section IV provides simulation results obtained using the RTDS. Conclusions are given in Section V.

II. MPC ALGORITHM

In this paper, a three-phase MMC topology will be analyzed. The three-phase MMC shown in Fig. 1 has three legs, each belonging to one of the three phases $j \in \{a, b, c\}$. Each leg consists of the upper and lower arms (denoted U and L, respectively). Each arm of the MMC comprises N_{SM} half-bridge submodules (SM), represented by their averaged equivalents, with R_{arm} and L_{arm} being the equivalent arm resistance and inductance, respectively. Furthermore, each SM contains a capacitor C_{SM}. The converter model is developed using the $\Sigma - \Delta$ nomenclature; the AC variables in the upper and lower converter arms of an arbitrary phase j can be represented as [26]:

$$i_j^\Delta = i_j^U - i_j^L, \qquad i_j^\Sigma = \frac{i_j^U + i_j^L}{2},$$
$$v_{Mj}^\Delta = \frac{-v_{Mj}^U + v_{Mj}^L}{2}, \qquad v_{Mj}^\Sigma = v_{Mj}^U + v_{Mj}^L, \tag{1}$$

where i_j^U and i_j^L are currents of the upper and lower arms, while v_{Mj}^U and v_{Mj}^L are the corresponding arm voltages. The DC and circulating current models are given with

$$\mathrm{d}i_{dc}t = \frac{1}{2L_{arm}}\left[3v_{dc} - \sum_{j=a,b,c} v_{Mj}^\Sigma - 2R_{arm}i_{dc}\right], \tag{2}$$

$$\mathrm{d}\mathbf{i_{jz}}t = \frac{1}{6L_{arm}}\left[\sum_{j=a,b,c} v_j^\Sigma - 3\mathbf{v_j^\Sigma} - 6R_{arm}\mathbf{i_{jz}}\right]. \tag{3}$$

The AC current is characterized by the following differential equation [2]:

$$\mathrm{d}i_j^\Delta t = \frac{2}{L_{arm} + 2L_r}\left[v_{Mj}^\Delta - \frac{R_{arm} + 2R_r}{2} i_j^\Delta\right], \tag{4}$$

where R_r and L_r are the respective values of phase resistance and inductance of the grid-side filter connected to the AC terminals of the MMC (Fig. 1).

The applied MPC method is described in detail in a recent publication [2]. The method is illustrated by the block diagram in Fig. 2. The DC-bus current and the circulating currents in all three phases are controlled using an optimal voltage level (OVL) MPC algorithm. The predicted values of currents for the next switching sample are derived from dynamic models of DC-bus and circulating currents. Zero-order discretization is applied when performing current prediction; that is, a constant voltage value is assumed during the entire switching period [2]. This means that the average voltage value over the observed switching period is used to estimate the current response. The prediction of DC-bus and circulating current values over the course of one switching interval T_{sw} is performed as follows:

$$\begin{aligned} i_{dc}^p\big((k+1)\big) &= i_{dc}(k)e^{-T_{sw}/\tau_z} \\ &\quad + \left(3V_{dc} - \sum_{j=a,b,c} V_{Mj}^\Sigma(k)\right)\frac{1 - e^{-T_{sw}/\tau_z}}{R_z}, \end{aligned} \tag{5}$$

$$\begin{aligned} \mathbf{i_{jz}^p}\big((k+1)\big) &= \mathbf{i_{jz}}(k)e^{-T_{sw}/\tau_z} \\ &\quad + \left(\sum_{j=a,b,c} V_{Mj}^\Sigma(k) - 3\mathbf{V_{Mj}^\Sigma}(k)\right)\frac{1 - e^{-T_{sw}/\tau_z}}{3R_z}, \end{aligned} \tag{6}$$

where $V_{Mj}^\Sigma(k) = N_j^\Sigma V_{SMj}(k)$, while $V_{SMj}(k)$ is the average submodule voltage across leg j at the time instant kT_{sw}. Parameters of the circulating current model are defined as $R_z = 2R_{arm}$ and $\tau_z = L_{arm}/R_{arm}$. Currents $i_{dc}^p(k+1)$ and $\mathbf{i_{jz}^p}(k+1)$ are predicted values at the end of the current switching period, while $i_{dc}(k)$ and $\mathbf{i_{jz}}(k)$ are measured DC and circulating current values at the beginning of the current switching period, in time instant kT_{sw}. The optimal number of submodules inserted in the observed leg, N_j^Σ, is obtained by minimizing the deviation between the reference values of DC and circulating currents and their respective responses, as described in detail in [2]. After obtaining the optimal value of N_j^Σ, the optimal number of submodules inserted into the upper arm of each phase N_j^U is determined. First,

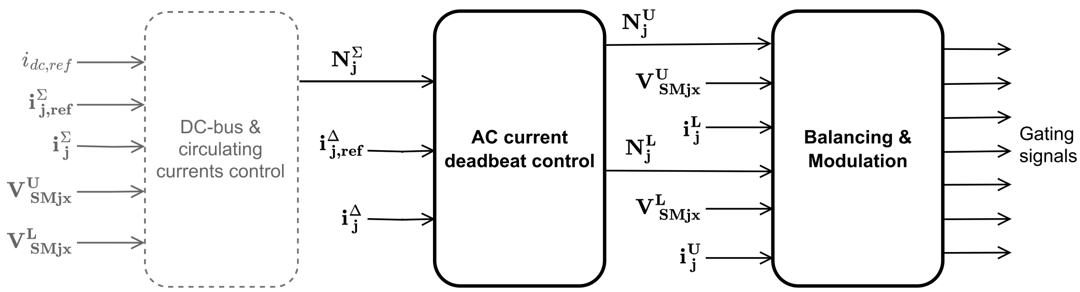


FIGURE 2. **Block diagram of the MPC method [2] (highlighted blocks are modified in this paper).**

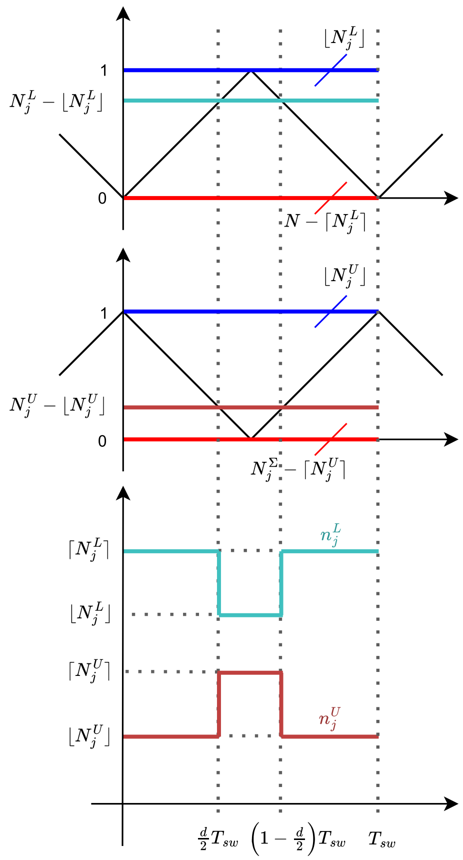


FIGURE 3. **Modulation principle used for MPC method; n_j^U and n_j^L represent instantaneous values of the number of inserted submodules in the upper and lower arms, respectively.**

the AC current equation (4) is discretized and rearranged to obtain the following form:

$$i_j^{\Delta,p}(k+1) = i_j^{\Delta}(k)e^{\frac{-T_{sw}}{\tau_{ac}}} + \frac{V_{Mj}^{\Delta}(k)}{R_{ac}}\left(1 - e^{\frac{-T_{sw}}{\tau_{ac}}}\right), \tag{7}$$

where $\tau_{ac} = L_{ac}/R_{ac}$ is the time constant, $L_{ac} = L_{arm}/2 + L_r$ and $R_{ac} = R_{arm}/2 + R_r$ are the equivalent AC inductance and resistance, respectively. The AC voltage $V_{Mj}^{\Delta}(k)$ can be expressed as

$$V_{Mj}^{\Delta}(k) = \frac{N_j^L(k)V_{SMj}^L(k) - N_j^U(k)V_{SMj}^U(k)}{2}, \tag{8}$$

where $V_{SMj}^U(k)$ and $V_{SMj}^L(k)$ are the average voltages of the submodules comprising the upper and lower arms, while $N_j^U(k)$ and $N_j^L(k)$ represent the number of submodules inserted in the upper and lower arms of the observed phase [2]. It should also be noted that the condition $N_j^U(k) + N_j^L(k) = N_j^{\Sigma}$ must be satisfied at all times. It should be noted that, while N_j^U and N_j^L are generally non-integer values, N_j^{Σ} is always an integer. To retain the simplicity of the algorithm, $V_{Mj}^{\Delta}(k)$ in (7) is assumed constant during the switching period. By replacing $i_j^{\Delta,p}(k+1)$ with the desired current reference $i_{j,ref}^{\Delta}(k+1)$, the number of inserted submodules in the upper arm is obtained as [2]

$$N_j^U(k) = \frac{N_j^{\Sigma}(k)V_{SMj}^L(k) - \frac{2R_{ac}\left(i_{j,ref}^{\Delta}(k+1) - i_j^{\Delta}(k)e^{-T_{sw}/\tau_{ac}}\right)}{1-e^{-T_{sw}/\tau_{ac}}}}{V_{SMj}^U(k) + V_{SMj}^L(k)}. \tag{9}$$

If the value N_j^U calculated from (9) is non-integer, the number of inserted submodules in the upper arm varies during the switching period, as shown in Fig. 3 and elaborated in [2]. The average value of the variable number of inserted submodules n_j^U over one switching period is equal to the requested non-integer value N_j^U. The instantaneous number of submodules inserted in the lower arm (n_j^L) is then varied accordingly to maintain a constant value of the submodules inserted per leg at all times ($n_j^U + n_j^L = N_j^{\Sigma}$). The varying number of submodules inserted during a single switching period clearly contradicts the assumption of constant AC voltage. This voltage variation may compromise algorithm accuracy under certain conditions, as explained shortly.

Provided that $\tau_{ac} \gg T_{sw}$, the current response changes linearly during the switching period, as observed in Fig. 4. The reference current waveform, if assumed sinusoidal, is also approximately linear over this interval. In that case, a triangular modulation carrier [2] ensures a close match between the current response and the reference at time instants $T_{sw}/2$ and T_{sw}, significantly improving the control accuracy. This justifies the previously mentioned application of average voltage in predicting the current response. However, if the

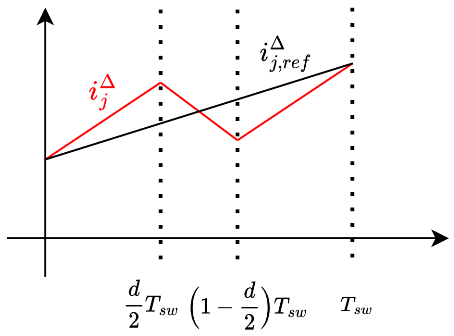


FIGURE 4. The AC current variation during one switching period in the case when $\tau_{ac} \gg T_{sw}$.

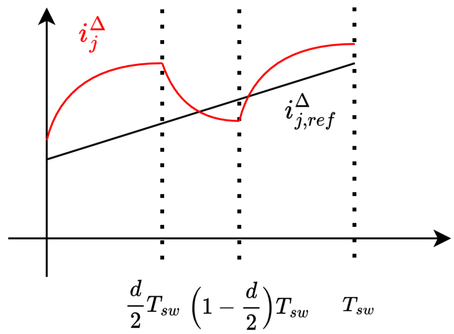


FIGURE 5. The AC current variation during one switching period when the condition $\tau_{ac} \gg T_{sw}$ is not satisfied.

share of R_{ac} in the AC impedance is increased, the current becomes appreciably non-linear and the response begins to deviate significantly from the reference, as shown in Fig. 5. In that case, the current response cannot be estimated using the average voltage value, as the superposition principle is no longer valid. Consequently, it is not possible to accurately determine the required number of inserted submodules using (9) in such scenarios.

Due to the non-linear current variation in the case where the filter resistance is significant, along with the fact that the voltage level changes twice during the switching period, the underlying approximations of the MPC algorithm become unacceptable. Obtaining the AC voltage reference that ensures a minimal current tracking error under such conditions becomes a task that the MPC algorithm in its current form is not capable of tackling. To solve this problem, the deadbeat algorithm is supplemented by an artificial neural network (NN) to improve AC current control in cases where the AC impedance contains a considerable resistive component.

III. NN-ASSISTED MPC ALGORITHM

Based on the discussion in the previous section, the MPC algorithm needs to be modified or supplemented to handle AC current control when the resistive component of the AC side impedance is significant. To address this issue, an NN is introduced in combination with the MPC framework. Guidelines for NN training and implementation in a high-fidelity real-time simulator are given in [4]. This approach results in a novel hybrid controller that leverages the strengths of MPC and NN, and is available as an open source online library [27], [28]. The NN is used to adjust the voltage reference, effectively improving the accuracy of the current response. The following subsections detail the development and structure of the proposed enhanced algorithm and its implementation.

The NN structure is rather conventional, with three layers: input, hidden, and output. The outputs of the NN are reference voltage increments resulting in improved current control. Further details on NN implementation are given in Subsection B. The most challenging aspect is data preparation, as the NN's outputs are initially unknown. Since these should align with reference voltage adjustments that lead to optimal current control performance, an optimization procedure must be carried out first. This process is elaborated in detail in Section A.

A. CONTROL OPTIMIZATION

The control variables in the MMC current control algorithm are the reference output voltages. More specifically, the number of submodules inserted per arm is determined. To improve current control, this number should be modified in each switching period by an unknown value. A switching period comprises two different switching states per phase. Each switching state is active during a predetermined time interval, thus obtaining the desired output voltage value.

An optimization procedure outlined in Algorithm 1 is developed and executed for each switching period. The optimization goal is to determine the optimal voltage reference values for each phase. The optimization criterion is to minimize the mean deviation between the AC current response and the reference over the switching period. The voltage reference $V_{j,ref}$ obtained from the MPC algorithm is modified by adding a voltage increment ΔV_j for each phase $j \in \{a, b, c\}$. These increments are optimized for each switching period to minimize the cost function J calculated as:

$$J = \max_{j} \left(\frac{1}{N} \left| \sum_{n=1}^{N} I_j(n) - I_{j,ref}(n) \right| \right), \tag{10}$$

where N is the number of samples per switching period, and $I_j(n)$ and $I_{j,ref}(n)$ are n^{th} samples of the measured and reference currents of phase j, respectively. It should be noted that $n \in \mathbb{N}$ is the sample number within the observed switching period. Each sample of the current response I_j is calculated based on the model with predetermined parameters and the modified reference voltage:

$$I_j(n+1) = I_j(n)\left(1 - \frac{T_s}{\tau_{ac}}\right) + V_j^*(n)\frac{T_s}{\tau_{ac}}, \tag{11}$$

where T_s is the sampling period and $V_j^*(n)$ is the n-th sample of the new applied voltage. The linearized form of (11) is justified by the fact that $T_s \ll \tau_{ac}$. To improve current

Algorithm 1 ***p*** **-Best Adaptive Differential Evolution for MMC Current Control Optimization**

Input: Simulation data for each phase $j \in \{a, b, c\}$: reference currents $I_{j,ref}$, measured currents I_j, reference voltages $V_{j,ref}$, model parameters.
Output: Optimized reference voltage increments for each phase ΔV_j.
Step 1: Initialize the population of candidate solutions with random voltage reference increments for each phase ΔV_j.
Step 2: Define the cost function J as the maximal mean deviation between the measured and reference currents according to (10).
Step 3: For each iteration i of the optimization perform:
for each candidate solution $\Delta V_j^{(i)}$ **do**
  Generate trial solution using p-best differential evolution with adaptive mutation and crossover.
  Evaluate the cost function J for the current solution by simulating the current waveform based on the model.
  if cost function J of trial solution is lower than the current solution **then**
    Accept the trial solution as the new solution.
  end if
end for
Step 5: Output optimized reference voltage increments ΔV_j.

tracking accuracy, i.e., minimize the cost function J, the reference voltage of each phase is modified as follows:

$$V_{j,ref}^* = V_{j,ref} + \Delta V_j, \tag{12}$$

where $V_{j,ref}$ is the original reference voltage obtained from the deadbeat algorithm and ΔV_j is the introduced reference voltage increment. Adding this increment results in modified applied voltage waveform V_j^* of each phase as seen in (11).

The unknown quantities in (11) and (12) are defined in Algorithm 1. It should be noted that the currents $I_{j,ref}$ and I_j in (10) and (11) correspond to $i_{j,ref}^{\Delta}$ and i_j^{Δ} in (9). The output of the optimization procedure is a matrix of reference voltage increments ΔV_j for all phases and switching periods. Based on the optimized data, the NN training and implementation process is performed as described in the following subsection. The initial sample of the current response $I_j(1)$ is not affected by the voltage modification and therefore corresponds to the value obtained from the simulation data.

B. NN DESIGN AND IMPLEMENTATION

The applied NN is a feedforward artificial neural network based on the library presented in [4] and consists of three layers. The NN structure is depicted in Fig. 6 and consists of the following layers:

- **Input Layer (6 neurons):** The input variables are reference currents $I_{a,ref}$ and $I_{b,ref}$; measured currents I_a and I_b; reference voltages (obtained from the MPC algorithm) $V_{a,ref}$ and $V_{b,ref}$.

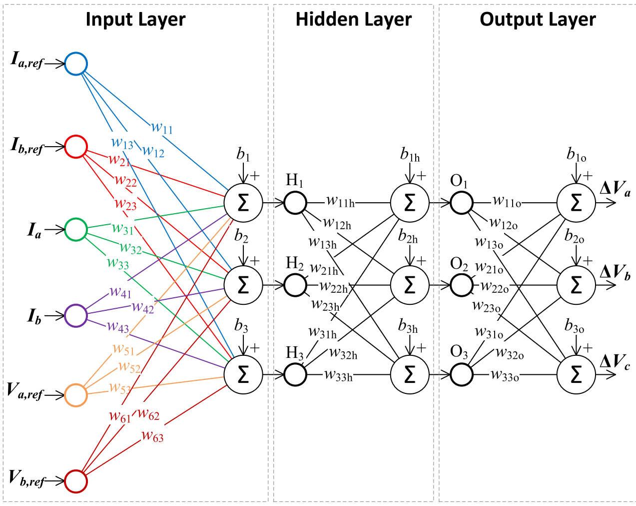


FIGURE 6. Neural network structure.

- **Hidden Layer (3 neurons):** Processing input features through learned weights and activation functions to capture complex relationships.
- **Output Layer (3 neurons):** The output variables are reference voltage increments ΔV_a, ΔV_b, and ΔV_c.

The values of NN weights and biases denoted in Fig. 6 can be obtained from [28]. Moreover, using the software available in [28], parameter values can be recalculated for different NN structures and system parameters.

Linear activation functions are used in all layers. The NN is trained to improve the current control of the MMC by computing reference voltage increments in each switching period. The increments are calculated based on the current reference and measured values and the reference voltage values obtained from the MPC algorithm outlined in Section II. The data is normalized by subtracting the mean of the dataset and dividing by its standard deviation for each value. The voltage increments, serving as target outputs, are derived from the control optimization described in the previous subsection.

IV. RESULTS

The performance of the proposed algorithm is evaluated in several scenarios for the MMC-based power system. Steady-state performance is evaluated first, after which the response to a varying current reference is observed. The results are obtained from high-fidelity real-time simulations performed using a state-of-the-art RTDS simulator [28]. The complete control algorithm is implemented on two processor cores operating at 3.5 GHz. The deadbeat algorithm and the NN are executed on a single core, running on a substep of 50 μs. The NN itself requires a total of 141 floating-point operations and 57 double-precision parameters. The values of the system parameters are shown in Table 2. Note that the sampling period of $T_s = 10$ μs results in $N = 100$ samples per switching period.

TABLE 2. System parameters.

Parameter	Designation	Value
DC link voltage	V_{dc}	525 kV
No. of submodules per arm	N_{sm}	200
MMC arm inductance	L_{arm}	40 mH
MMC arm resistance	R_{arm}	78.5 mΩ
Fundamental frequency	f	50 Hz
Load inductance	L_r	1 µH
Load resistance	R_r	100 Ω
Switching frequency	f_{sw}	1 kHz
Sampling period	T_s	10 µs

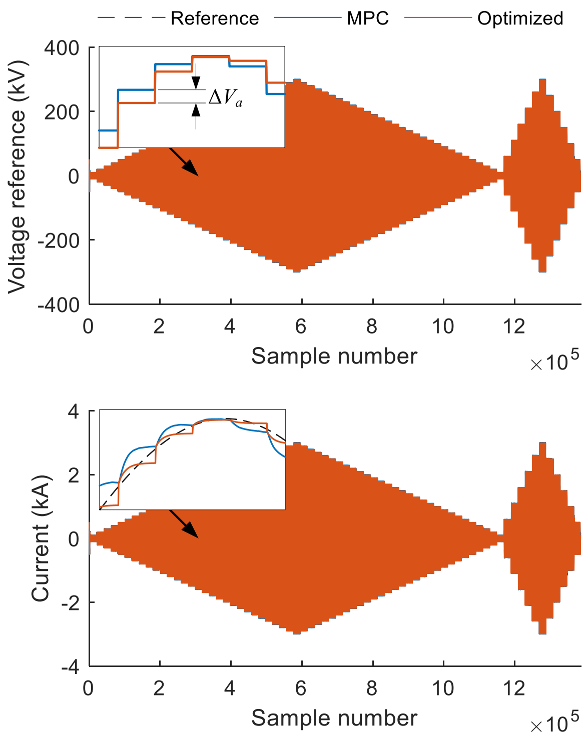
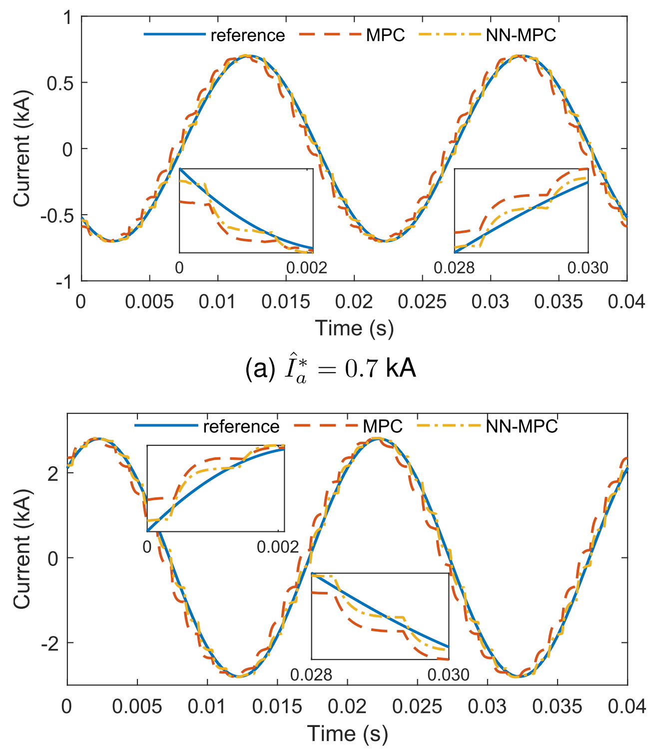


FIGURE 7. Optimization data set; the enlarged details show the optimized voltage increments and the corresponding current response.



(a) $\hat{I}_a^* = 0.7$ kA

(b) $\hat{I}_a^* = 2.8$ kA

FIGURE 8. Comparison of steady-state performance of the MPC and NN-MPC algorithms for two different values of reference current magnitudes.

The optimization procedure described in III-A was carried out for a total of 13,858 switching periods. The current reference magnitude was varied between 0 and 3 kA, achieving a wide range of operating modes and resulting in a large and diverse database for the NN design. The cost function J is reduced by 92% on average by introducing optimized reference voltage variations. The current tracking error is reduced by 90% or better in 86% of the analyzed PWM periods, whereas the control performance is deteriorated only in 10 cases. This deterioration can be attributed to the stochastic nature of the optimization procedure. Since the differential evolution algorithm is executed with a finite number of iterations, convergence to the global optimum is not guaranteed in every switching period, which results in suboptimal voltage increments in rare cases. The database used in the NN design process includes all PWM periods with J reduced by 80% or more, for a total of 13,650 viable data points. The phase a voltage and current data sets are shown in Fig. 7. The blue waveforms labeled "MPC" correspond to the optimization input data. The orange waveforms labeled "Optimized" correspond to the optimized voltage and current waveforms. The voltage increment and the resulting improved current response are denoted in the enlarged details of the respective diagrams.

The optimized data is used to train the NN. The data is first divided into training (90%) and testing (10%) sets. The training set is further divided into training and validation data with an 80:20 ratio. The validation split is used for monitoring NN performance during training. The network achieves a loss of $2.58 \cdot 10^{-4}$, which is a very good value considering that normalized data is used. The implementation of NN on RTDS requires normalization of the input data and denormalization of the output data (voltage increments). The voltage references are modified in each switching period by adding the increments obtained as the NN outputs to the values obtained from the MPC. Modified voltage references are included in the PWM modulation described briefly in Section II, with more details available in [2]. It should be noted that NN execution has low computational impact, as the MPC algorithm is significantly more complex.

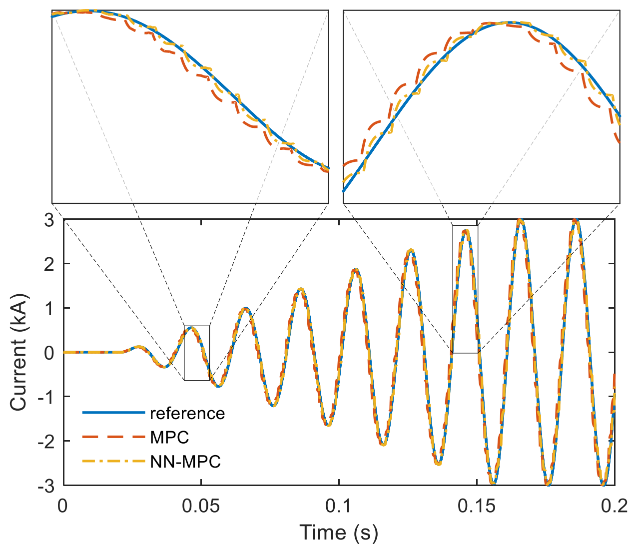


FIGURE 9. Comparison of transient performance of MPC and NN-MPC.

Fig. 8 shows the simulation results obtained in two steady-state operating modes with different reference current magnitudes. The enlarged details in the figure reveal that the proposed NN-assisted MPC algorithm outperforms the existing algorithm in both cases. The current waveform obtained without employing the NN exhibits a notable shift with respect to the reference signal, mostly due to the resistive component of the AC side impedance. The time-varying reference tracking capability of the proposed algorithm is demonstrated in Fig. 9. As seen in the enlarged details, the NN-MPC algorithm ensures a more accurate response compared to the MPC, with no noticeable delay.

Although the results displayed in Figs. 8 and 9 visually confirm the benefits of the proposed control approach, numerical performance indicators are required to further validate the performance improvement achieved by implementing the NN-MPC algorithm. Three commonly used indicators are selected for this purpose: the cross-correlation coefficient, the sum of squared errors, and the estimated delay. The values of these indicators are calculated for ten steady-state operating modes and the transient corresponding to Fig. 9. Each steady-state condition is characterized by a different reference current magnitude, ranging between 0.1 kA (mode 1) to 2.8 kA (mode 10) with a 0.3 kA step.

The results of Table 3 demonstrate the advantages of NN-MPC. The cross-correlation coefficient (CCR) is calculated using the Matlab function `xcorr`. CCR quantifies the similarity between two waveforms (reference and measured current), with values close to unity indicating a good match. Both algorithms exhibit good performance; however, NN-MPC is superior in all observed cases. Similar CCR values in all cases demonstrate consistent performance of both algorithms.

Although the CCR value is higher when NN-MPC is used, this is not sufficient for a conclusive evaluation. Therefore, additional indicators, time shift and the sum of squared errors, are calculated to further demonstrate the benefits of the proposed approach. The time shift is estimated using the Matlab function `finddelay`, which determines the number of samples by which one signal should be shifted with respect to the other to achieve the maximum value of the CCR. The number of samples is multiplied by the sampling period T_s to obtain the corresponding time value. According to Table 3, employing the NN-MPC algorithm significantly reduces the time shift. This confirms and generalizes observations from Figs. 8 and 9.

TABLE 3. Performance indicators of the MPC and NN-MPC algorithms.

Mode	CCR		Time shift (μs)		SSE	
	MPC	NN-MPC	MPC	NN-MPC	MPC	NN-MPC
1	0.9864	0.9979	490	20	270.8	41.79
2	0.9869	0.9985	470	70	261.4	30.19
3	0.9869	0.9985	410	30	261.7	29.53
4	0.9870	0.9986	470	30	259.0	29.26
5	0.9870	0.9986	440	80	258.2	29.03
6	0.9871	0.9986	490	20	256.1	28.62
7	0.9873	0.9986	460	0	252.5	27.94
8	0.9875	0.9987	480	0	248.4	27.21
9	0.9878	0.9987	380	10	242.5	26.23
10	0.9882	0.9988	460	0	234.7	25.21
Trans	0.9883	0.9987	460	50	104.0	11.95

Finally, the sum of squared errors (SSE) is determined as another measure of similarity, or rather difference, between two waveforms. The SSE is calculated using the Matlab function `sse` and normalized for the reference current magnitude in each operating mode. Lower values indicate better performance, therefore, the NN-MPC once again proves to be a significant improvement compared to the MPC algorithm in all analyzed cases.

It should be noted that the time shift and high SSE are not the consequence of algorithm latency, but can be attributed mainly to the deadbeat algorithm itself. Firstly, the AC current prediction is performed using the approximate expression (7), resulting in a mismatch between the obtained and reference current at the end of each switching period. Secondly, and arguably more importantly, the very principle of deadbeat control causes a significant deviation between the obtained and reference current waveforms when the resistive component of AC impedance is significant. This deviation is manifested as the exists even if deadbeat operates with no error, which raises the need for employing the NN.

V. CONCLUSION

This paper demonstrates and improved MPC deadbeat current control algorithm capable of handling arbitrary loads. The MPC deadbeat algorithm demonstrates strong performance in minimizing the THD of the AC current when the AC terminal impedance of the MMC is highly inductive.

However, when the resistance becomes comparable to the reactance, reference tracking becomes more challenging. This paper has shown that incorporating a neural network can improve the accuracy of AC reference current tracking in such cases by minimizing the mean deviation between the reference and the actual response. Furthermore, the simultaneous operation of the MPC deadbeat algorithm and the NN allows for a lower-complexity NN architecture while maintaining effective control performance, with the MPC deadbeat algorithm remaining the primary control method. Other potential benefits of introducing neural networks into the control algorithm, such as reduced model dependency and adaptive control capability under parametric uncertainty are yet to be explored. Future work will address the integration of the NN-MPC-controlled MMC within a larger network and assess its performance under large-signal disturbances.

REFERENCES

[1] L. A. M. Barros, A. P. Martins, and J. G. Pinto, "A comprehensive review on modular multilevel converters, submodule topologies, and modulation techniques," *Energies*, vol. 15, no. 3, 2022.

[2] M. Majstorović, V. Nougain, L. Ristić, and A. Lekić, "Deadbeat-based control for MMC-HVDC power systems," *Int. J. Electr. Power Energy Syst.*, vol. 167, Jun. 2025, Art. no. 110583.

[3] L. Liu, A. D. Shetgaonkar, and A. Lekic, "Interoperability of classical and advanced controllers in MMC based MTDC power system," *Int. J. Electr. Power Energy Syst.*, vol. 148, Jun. 2023, Art. no. 108980.

[4] B. Masalmeh, R. Prasad, V. Nougain, and A. Lekic, "Neural networks in RSCAD: Enhancing MMC-based HVDC simulation with advanced machine learning," *IEEE Trans. Ind. Appl.*, vol. 61, no. 4, pp. 1–10, Jul. 2025.

[5] D. Wang et al., "Model predictive control using artificial neural network for power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3689–3699, Apr. 2022.

[6] S. Wang, T. Dragicevic, Y. Gao, and R. Teodorescu, "Neural network based model predictive controllers for modular multilevel converters," *IEEE Trans. Energy Convers.*, vol. 36, no. 2, pp. 1562–1571, Jun. 2021.

[7] R. Mourouvin et al., "Real-time hierarchical MPC applied to an MMC in grid-forming mode: Implementation and validation in power hardware-in-the-loop," *IEEE Trans. Power Del.*, vol. 38, no. 5, pp. 3142–3154, Oct. 2023.

[8] A. Shetgaonkar, L. Liu, A. Lekic, M. Popov, and P. Palensky, "Model predictive control and protection of MMC-based mtdc power systems," *Int. J. Electr. Power Energy Syst.*, vol. 146, Mar. 2023, Art. no. 108710. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061522007062

[9] X. Liu et al., "Predictor-based neural network finite-set predictive control for modular multilevel converter," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11621–11627, Nov. 2021.

[10] X. Liu et al., "Data-driven neural predictors-based robust MPC for power converters," *IEEE Trans. Power Electron.*, vol. 37, no. 10, pp. 11650–11661, Oct. 2022.

[11] W. Wu, L. Qiu, X. Liu, J. Ma, J. Rodríguez, and Y. Fang, "Data-enabled finite state predictive control for power converters via adaline neural network," *IEEE Trans. Ind. Electron.*, vol. 72, no. 3, pp. 2244–2253, Mar. 2025.

[12] X. Liu, L. Qiu, Y. Fang, and J. Rodríguez, "Predictor-based data-driven model-free adaptive predictive control of power converters using machine learning," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 7591–7603, Aug. 2023.

[13] S. Sabzevari, R. Heydari, M. Mohiti, M. Savaghebi, and J. Rodriguez, "Model-free neural network-based predictive control for robust operation of power converters," *Energies*, vol. 14, no. 8, p. 2325, Apr. 2021.

[14] S. Wang, T. Dragicevic, G. F. Gontijo, S. K. Chaudhary, and R. Teodorescu, "Machine learning emulation of model predictive control for modular multilevel converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11628–11634, Nov. 2021.

[15] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicevic, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output *LC* filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019.

[16] S. Singh, D. M. Stipanovic, and A. Lekic, "Machine learning-driven model predictive control of modular multilevel converters," *IEEE Access*, vol. 13, pp. 169327–169339, 2025.

[17] N. Yousefi, J. Ebrahimi, and A. Bakhshai, "Neural network controller based on direct and indirect model predictive for modular multilevel converters," in *Proc. 25th Eur. Conf. Power Electron. Appl.*, 2023, pp. 1–8.

[18] A. Bakeer, I. S. Mohamed, P. B. Malidarreh, I. Hattabi, and L. Liu, "An artificial neural network-based model predictive control for three-phase flying capacitor multilevel inverter," *IEEE Access*, vol. 10, pp. 70305–70316, 2022.

[19] M. Novak and T. Dragicevic, "Supervised imitation learning of finite-set model predictive control systems for power electronics," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1717–1723, Feb. 2021.

[20] H. Wang, Y. Yue, B. Sun, and H. Zhao, "Neural network-based model predictive control approach for modular multilevel converters," in *Proc. 4th Int. Conf. Electron. Commun. Artif. Intell. (ICECAI)*, May 2023, pp. 324–330.

[21] F. Simonetti, A. D'Innocenzo, and C. Cecati, "'Neural network model predictive control for CHB converters with FPGA implementation," *IEEE Trans. Ind. Informat.*, vol. 19, pp. 9691–9702, 2023.

[22] Z. Gong, P. Tuo, C. Zheng, and X. Wu, "Design and analysis of the model predictive control implemented by the ANN technique for MMC-based rectifier with improved grid adaptability," *IEEE Trans. Power Electron.*, vol. 39, no. 10, pp. 12306–12322, Oct. 2024.

[23] W. Zhang, J. Zhang, Q. Wang, Y. Jiang, and G. Tan, "Dual-vector model predictive control for modular multilevel converter with low calculation burden," *IEEE Access*, vol. 12, pp. 28520–28530, 2024.

[24] J. Wang, Y. Tang, P. Lin, X. Liu, and J. Pou, "Deadbeat predictive current control for modular multilevel converters with enhanced steady-state performance and stability," *IEEE Trans. Power Electron.*, vol. 35, no. 7, pp. 6878–6894, Jul. 2020.

[25] J. Wang, Y. Tang, Y. Qi, P. Lin, and Z. Zhang, "A unified startup strategy for modular multilevel converters with deadbeat predictive current control," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 6401–6411, Aug. 2021.

[26] A. Shetgaonkar, T. Karmokar, M. Popov, and A. Lekic, "Enhanced real-time multi-terminal HVDC power system benchmark models with performance evaluation strategies," *CIGRE Sci. Eng.*, vol. 32, pp. 1–29, 2024.

[27] M. Majstorovic, B. Brkovic, V. Nougain, L. Ristic, and A. Lekic, "Deadbeat-based control for MMC-HVDC power systems with and without adaptiveness (version v1.0)," Tech. Rep.

[28] *Rtds Models for Deadbeat-Based Control for MMC-HVDC Power Systems With and Without Adaptiveness*. Accessed: Mar. 26, 2026. [Online]. Available: https://github.com/control-protection-grids-tudelft/SUNRISE_MPC.git