IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 October 2022, accepted 13 December 2022, date of publication 19 December 2022,
date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230356

RESEARCH ARTICLE

Low Complexity MPC-DSVPWM for Current Control of PMSM Using Neural Network Approach

HASAN ALI GAMAL AL-KAF, (Graduate Student Member, IEEE), AND KYO-BEUM LEE, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, South Korea

Corresponding author: Kyo-Beum Lee (kyl@ajou.ac.kr)

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea under Grant 20206910100160 and Grant 20225500000110.

ABSTRACT Multilayer neural network-based model predictive control (MLNN-MPC) has received a lot of attention in different power electronic applications. However, the computational burden often imposes limitations in low-order DSPs especially if a large number of voltage vectors (VVs) are used. The execution time of MLNN-MPC in low-order DSPs is affected heavily by the number of input, output, neurons in the hidden layer, and the type of activation function. Furthermore, MLNN contains many parameters that needed to be optimized, such as initial weights, number of iterations, and number of neurons. Therefore, in this study, a creative single-layer neural network-based model predictive control with discrete space vector PWM (SLNN-MPC-DSVPWM) is proposed to overcome these limitations. The main advantages of the proposed method include easy implementation on low-order DSPs, better performance compared with MLNN-MPC, allowing the use of a large number of VVs, and no initialization of lookup tables for all VVs. The proposed SLNN is trained using the Levenberg-Marquardt algorithm and results in an execution time of only 8 μs compared with the complexity of the conventional MPC-DSVPWM and recent MLNN-MPC methods. The SLNN-MPC-DSVPWM is validated by both simulation and experimental results for permanent magnet synchronous motors.

INDEX TERMS Artificial neural network (ANN), discrete space vector pulse width modulation (DSVPWM), execution time, model predictive control (MPC), permanent magnet synchronous motor (PMSM).

I. INTRODUCTION

Permanent magnet synchronous motors (PMSMs) have recently gained popularity, particularly in variable-speed drives, owing to their high torque density and efficiency [1]. Field-oriented control (FOC) and direct torque control (DTC) are the most popular control strategies used in PMSM drives. FOC has better steady-state performance; however, it requires coordinate transformation, fine-tuning for the PI controller, and pulse with modulation (PWM) generation. In contrast, DTC has a fast dynamic response and simple control structure compared to the FOC, which is preferred for high dynamic applications. However, it produces variable switching frequency and large torque ripple [2], [3].

Owing to the significant development of fast microprocessors, model predictive control (MPC) has recently attracted considerable attention. This can be attributed to its ease of implementation, the inclusion of nonlinearity, multivariable control, and fast dynamics [4], [5], [6], [7], [8], [9], [10], [11]. Despite the advantages of MPC, improving its performance and calculation processing are regarded as key concerns in industrial applications. Conventional MPC has a variable switching frequency, which results in large current ripples with a wide harmonic spectrum. Large current ripples are

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

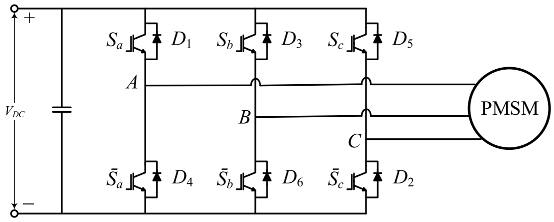


FIGURE 1. Two-level voltage-source inverter connected with PMSM.

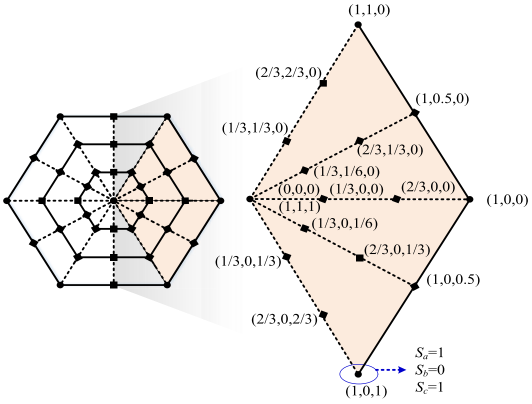


FIGURE 2. Space vector diagram with 37 VVs.

common when single actual voltage vector (VV) is applied per discrete sample period (T_s) [12].

Many studies have been conducted to improve the performance of conventional MPCs. One approach is to apply two or more VVs per discrete sample period by calculating the vector duty ratio [13], [14], [15]. The main difficulty of this method is the generation of suitable VVs that produce lower current, torque, and stator flux ripples. In addition, variable switching frequency and high computational burden are the main drawbacks of this approach [16]. Another method implements discrete space vector modulation (DSVM) with MPC to synthesize a wide number of virtual VVs within a T_s [17], [18], [19]. The cost function determines the best VV among real, virtual, and zero VVs and is applied to the converter through SVM. A merit of using this approach is that the steady-state performance can be improved by achieving a constant switching frequency. However, the high computational burden and the need for the initialization of lookup tables for a large number of virtual VVs are the limitations of this approach, especially in low-speed/order microprocessors.

The issue of the computational burden of MPC-DSVM has been solved recently. The main solution is using a deadbeat approach [20], [21]. The concept of the deadbeat control is to use a mathematical model to derive a reference which finds the nearest VV that minimizes the cost function. In [20], the deadbeat method has been implemented for predictive torque control (PTC) that utilizes only three VVs and does not require a predefined lookup table. In [21], a robust deadbeat with an improved DSVM method is proposed. The improved DSVM selects the VVs located in a single sector and two concentric hexagonal diagrams. In addition, the proposed method uses a simple equation to generate the VVs without requiring lookup table initialization. Based on the same virtual VV generalization, another recent method uses a unidirectional VV preselection method for PTC of the PMSM [22]. The proposed method uses the stator flux sector and speed direction to reduce the flux and torque ripples. The prior studies are based on model prediction and cost function minimization, which do not change the main concept of MPC and require a longer execution time in high-order horizon lengths and multilevel power converters.

In recent years, artificial neural network (ANN) has emerged as an excellent approach to improve the control performance of MPC in power converters [23], [24]. Multilayer neural network based model predictive control (MLNN-MPC) is proposed to reduce the execution time of MPC. The main idea is to train the MLNN offline by collecting data from a conventional MPC. Once the MLNN can select the optimal VV, a simple mathematical equation of the MLNN can be used in real-time without the computational complexity of the cost function. MLNN-MPC has been implemented only for a small number of VVs and low-level topologies, including two-level inverters [25], [26], and modular multilevel converters [23]. Further investigation into the use of MLNN in a large number of VVs and complex multilevel converters is required. In [27], a MLNN-MPC was successfully implemented on a single-phase five-level flying capacitor converter and a three-phase three-level flying capacitor converter using an FPGA-based control system. However, the proposed method can only operate using high-speed microprocessors, such as FPGA. Such controllers can perform more than one operation concurrently and are suitable for high-speed sampling frequencies [28]. For low-speed sampling frequencies, such as the DSP control board, the MLNN proposed in previous studies required a large execution time, which limited its real-time implementation. This is due to the multiplication of two $n \times n$ matrices that performed *O(n3)* operations [29]. In addition, the execution time of MLNN-MPC is affected heavily by the number output which represents the number of VVs that is used, neurons the in hidden layer, and the type of activation function. If only one hidden layer is used, it results in a minimum of two matrix multiplications and two activation functions. Another drawback of the MLNN is that it contains many parameters that are needed to be optimized, such as initial weights, number of iterations, and number of neurons.

On the other hand, single-layer neural networks (SLNN) trained with Levenberg–Marquardt outperformed complex neural networks such as extreme learning machines (ELM) and MLNN in terms of predictive accuracy in a complex multivariable problem [30]. Based on the comprehensive analysis presented in the literature, MPC has not been accompanied by an ANN for motor drive appliances. Therefore, in this study, a SLNN with model predictive control with discrete space vector PWM (SLNN-MPC-DSVPWM) is proposed to overcome the computational problem of MLNN-MPC and conventional MPC-DSVPWM for a large number of VVs in a low-order DSP for PMSM motor drive applications. The SLNN is proposed to use a high computation activation function, such as sigmoid activation, in the training phase, and a low computation activation function, such as a linear activation function, in real-time DSP without degrading the performance.

Furthermore, the proposed SLNN does not require a table for VVs initiation, which is based on selecting a suitable value for each switch signal and implementing a mathematical equation to produce the required VVs using DSVPWM. Importantly, it needs to be mentioned that the proposed method is not intended to outperform the conventional MPC-DSVPWM and MLNN controllers. The main advantage of the proposed method is that its execution time is independent of the complexity and has the lowest execution time for low-order DSP. Therefore, the proposed method can be easily implemented for a large number of VVs to improve the performance of MPC while MPC-DSVPWM and MLNN suffer for high execution time.

The proposed SLNN-MPC-DSVPWM method was simulated and experimentally verified for a three-phase two-level inverter-fed PMSM drive. The followings are the main contribution of this paper.

1) It is first implemented for motor drive appliances.
2) It is implemented easily on low-order DSPs, with an execution time of 8 μs.
3) It is first implemented to solve the limitation of MPC-DSVPWM when a large number of VVs are used.
4) It does not require the initialization of lookup tables for all VVs generated by DSVPWM.
5) It is compared with recent MLNN-MPC and MPC-DSVPWM.

II. MPC-DSVPWM FOR PMSM

A. PMSM MODELING FOR MPC-DSVPWM

A PMSM fed by a two-level inverter is used to carry out all the necessary explanations in this study, as shown in Fig. 1. The voltage equations of PMSM in the reference frame of the rotor are shown in (1),

$$\begin{aligned} v_d &= R_s I_d + L_d \frac{d}{dt}\left(I_d\right) - \omega_e L_q I_q, \\ v_q &= R_s I_q + L_q \frac{d}{dt}\left(I_q\right) + \omega_e L_d I_d + \omega_e \phi_f, \end{aligned} \tag{1}$$

where v_d, v_q, I_d, and I_q are the voltage and current components of the direct (d) and quadrature (q) axes. R_s is the stator resistance per phase, L_d and L_q are the d- and q-axis inductances, ω_e is the electrical rotor speed, and φ_f is the flux linkage established by the rotor. To obtain the discrete-time model based on forward Euler discretization, the derivative dI/dt can be approximated as shown in (2):

$$\frac{d}{dt} = \frac{I_{(k+1)} - I_{(k)}}{T_s}, \tag{2}$$

where T_s denotes the sampling interval. The current prediction of PMSM can be calculated using (1) and (2) as in (3) [22],

$$\begin{cases} I^z_{\alpha(k+1)} = I_{\alpha(k)} + \dfrac{T_s}{L_d}\left(-R_s I_{\alpha(k)} + L_q \omega_e I_{\beta(k)} + v_{\alpha(k)}\right) \\ I^z_{\beta(k+1)} = I_{\beta(k)} - \dfrac{T_s}{L_q} \\ \qquad \left(R_s I_{\beta(k)} + L_d \omega_e I_{\alpha(k)} + L_q \omega_e \phi_f - v_{\beta(k)}\right) \end{cases} \tag{3}$$

where $I^z\alpha(k+1)$ and $I^z_\beta(k+1)$ are currents in the stationary reference frame. The cost function (g^z) for measuring the error between the reference and predicted currents to evaluate all the VVs generated by DSVPWM is as described in (4),

$$g^z = \left|I^*_{\alpha(k+1)} - I^z_{\alpha(k+1)}\right| + \left|I^*_{\beta(k+1)} - I^z_{\beta(k+1)}\right|. \tag{4}$$

DSVPWM allows $v_{\alpha(k)}$ and $v_{\beta(k)}$ to be located in any position of the inverter control region. Fig. 2 shows the control region of a two-level three-phase inverter, where the real and virtual VVs are presented in round and square marks, respectively. In this figure, the VVs are equally divided into three hexagons with 37 VVs. DSVPWM is used in this study instead of DSVM due to its low complexity and better performance owing to its natural ability in generating a constant switching frequency without sophisticated dwelling time calculating for each VV. [31]. Furthermore, it is easy to implement in most of DSPs due to the availability of the ePWM-predefined function. Moreover, SVPWM can improve modulation region by 15.5% compared to sinusoidal PWM [31]. $v_{\alpha(k)}$ and $v_{\beta(k)}$ are transferred into three-phase reference voltages v^*_A, v^*_B, and v^*_C using an inverse transformation. The injected offset voltage (v^*_{sn}) is calculated based on the principle of DSVPWM as in (5),

$$v^*_{sn} = \frac{-(v^*_{\max} + v^*_{\min})}{T_s} \tag{5}$$

where $v^*_{\max}$ and $v^*_{\min}$ denote the maximum and minimum of the v^*_A, v^*_B, and v^*_C, respectively [31]. To simplify the calculation process for generating the three-phase reference pole voltages, the A-phase reference pole voltage (v^*_{An}) is calculated as shown in (6). The same process can be applied to the other two phases (B and C) as well. As per the procedure, v^*_{sn} is added to v^*_A as expressed in (6),

$$v^*_{An} = v^*_A + v^*_{sn}. \tag{6}$$

Notably, the normalized A-reference signal can be generated using (7) as,

$$v^*_{Az} = \frac{(2 + v^*_{An})}{V_{dc}}, \tag{7}$$

where V_{dc} is the DC-link voltage. Meanwhile, v^*_{Az} is compared with a PWM triangular carrier V_{CA} to generate the

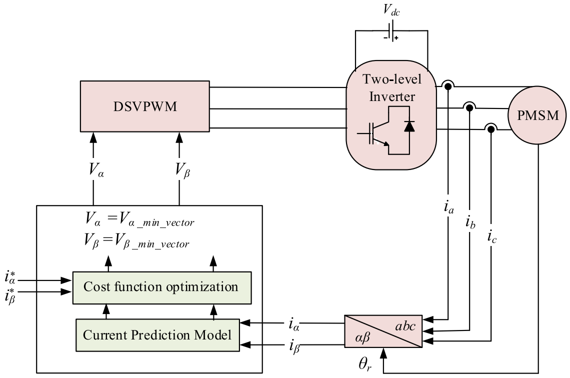


FIGURE 3. Control block diagram of the MPC-DSVPWM method.

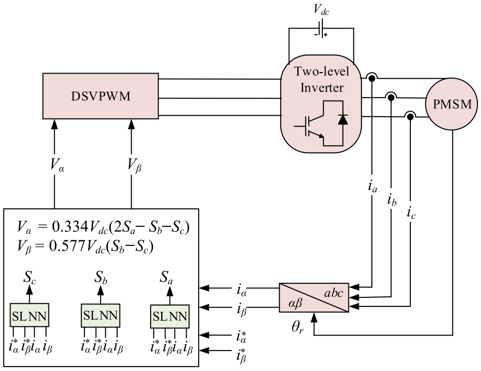

V_{dc}
Two-level Inverter
DSVPWM
PMSM
V_α
V_β
$V_\alpha = 0.334V_{dc}(2S_a - S_b - S_c)$
$V_\beta = 0.577V_{dc}(S_b - S_c)$
S_c
S_b
S_a
SLNN
SLNN
SLNN
$i_\alpha^* \, i_\beta^* \, i_\alpha \, i_\beta$
$i_\alpha^* \, i_\beta^* \, i_\alpha \, i_\beta$
$i_\alpha^* \, i_\beta^* \, i_\alpha \, i_\beta$
i_a
i_b
i_c
i_α
i_β
i_α^*
i_β^*
$\alpha\beta$
abc
θ_r

FIGURE 4. Control block diagram of the proposed SLNN-MPC- DSVPWM method.

switching signals, as shown in (8),

$$\begin{cases} v_{Az}^* \geq V_{CA}, S_a = 1; \overline{S_a} = 0 \\ v_{Az}^* < V_{CA}, S_a = 0; \overline{S_a} = 1, \end{cases} \tag{8}$$

where S_a and $\bar{S}_a$ are the inverter switching signals. A complete block diagram of the conventional MPC-DSVPWM is shown in Fig. 3.

III. PROPOSED SLNN-MPC-DSVPWM FOR PMSM

A. THE MAIN DIFFERENCES BETWEEN MLNN AND SLNN

MLNN has a hidden layer and each hidden layer has a number of neurons that need to be optimized, whereas the SLNN has a simpler structure where no hidden layers are required. Implementing SLNN and checking the accuracy of the SLNN is one technique to evaluate whether the problem is linear or not; if the SLNN gives high accuracy, the problem is linear and can be solved using SLNN. In addition, advanced algorithms such as Levenberg–Marquardt algorithm can improve the performance of SLNN and outperform MLNN and ELM [30].

B. PROPOSED SLNN-MPC-DSVPWM

Figs. 4 and 5 show the general block diagrams of the proposed SLNN-MPC-DSVPWM. Offline training of the SLNN was used to replace the heavy-duty computation of the current prediction model and cost function evaluation. The input and output of the MPC are collected for training the proposed SLNN using PSIM software. The input parameters are I_α, I_β, $I_\alpha*$, and $I_\beta*$ currents in a stationary reference frame, where the output is the optimal value of the switching signals S_a, S_b, and S_c. Then, using MATLAB software, the data collected is split into 70% for training and 30% for testing. The training of proposed SLNN is done offline to avoid the large computation time for training process. The proposed method comprises three SLNNs, and each SLNN selects a suitable switch state for each phase. Then by calculating the v_α and v_β as in (9) [31], any real and virtual voltage vector can be easily generated without the need for initialization for each VV. For example, if the proposed SLNN selects $S_a = 1$, $S_b = 0$, and $S_c = 0$ as optimal switching state, then by substituting the switching state in (9), the $v_\alpha = 2/3V_{DC}$ and $v_\beta = 0$, which represent V_1.

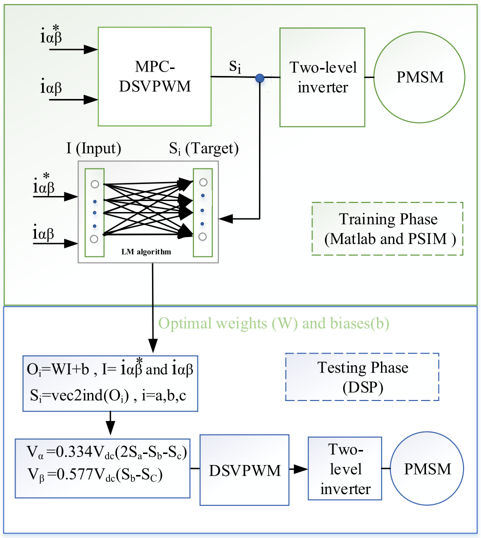

$i_{\alpha\beta}^*$
$i_{\alpha\beta}$
MPC-DSVPWM
S_i
Two-level inverter
PMSM
I (Input)
S_i (Target)
$i_{\alpha\beta}^*$
$i_{\alpha\beta}$
LM algorithm
Training Phase (Matlab and PSIM)
Optimal weights (W) and biases(b)
O_i=WI+b , I= $i_{\alpha\beta}^*$ and $i_{\alpha\beta}$
S_i=vec2ind(O_i) , i=a,b,c
Testing Phase (DSP)
$V_\alpha = 0.334V_{dc}(2S_a - S_b - S_c)$
$V_\beta = 0.577V_{dc}(S_b - S_C)$
DSVPWM
Two-level inverter
PMSM

FIGURE 5. Control block diagram of training and testing phases for the proposed SLNN-MPC-DSVPWM method.

$$\begin{aligned} v_\alpha &= \frac{V_{DC}}{3}(2S_a - S_b - S_c), \\ v_\beta &= \frac{V_{DC}}{\sqrt{3}}(S_b - S_c). \end{aligned} \tag{9}$$

Then v_α and v_β are transferred to v_A, v_B, and v_C using inverse transformation and are applied to the inverter through DSVPWM. The advantage of using three SLNNs is to avoid lookup table initialization for a large number of virtual VVs. In addition, this method can reduce the execution time of the SLNN-MPC-DSVPWM which does not require a large multiplication matrix of the input with all VVs. Instead, it simply multiplies the input with suitable switch states for each phase.

Fig. 6 shows the process of SLNN. Data collection and splitting of training and testing is the first step in any neural network model. The second step is building the neural network model, the initial weights (W) and biases (b) are set to zero, sigmoid function is chosen as activation function, the number of iterations is 10, and the Levenberg-Marquardt is chosen for training and updating the W and b. After the training process is finished, the optimal W and b are extracted and the linear activation function is used instead of the sigmoid activation function. Theses parameters are only used in real-time DSP to classify and choose the optimal switching state. A further discussion of the design and mathematical of the proposed SLNN is followed in the next section.

C. STRUCTURE OF SLNN

The mathematical representation of SLNN is expressed in (10),

$$\begin{aligned} H &= g(x), \\ x &= WI + b, \end{aligned} \tag{10}$$

where I is the input vector of the network, H is the output vector of the output layer, W is the weight matrix of the output layer, b is the bias vector, and g is a sigmoid function, which can be expressed by (11),

$$g = \frac{1}{1 + e^{-x}}. \tag{11}$$

However, the execution time of the sigmoid function is high when applied to a low-order DSP. Therefore, the proposed method uses the sigmoid function only in the training phase

Fast sigmoid and linear activation functions have lower execution times, as described in (12) and (13), respectively,

$$g = \frac{1}{1 + |x|}, \tag{12}$$

$$g = x. \tag{13}$$

Fig. 7 show the structure of the SLNN during the training and testing phases. In the training phase, the sigmoid function is used to achieve good input-output mapping. In the testing phase, optimal W and b are used with a linear activation function to obtain the lowest execution time. Advanced algorithms can improve the SLNN and outperform MLNN and ELM. The Levenberg–Marquardt algorithm is used to update the weights, as shown in (14),

$$\begin{aligned} W(t+1) &= W(t) - \Delta W(t), \\ \Delta W(t) &= (JJ^t + \mu I)^{-1} J^t e, \end{aligned} \tag{14}$$

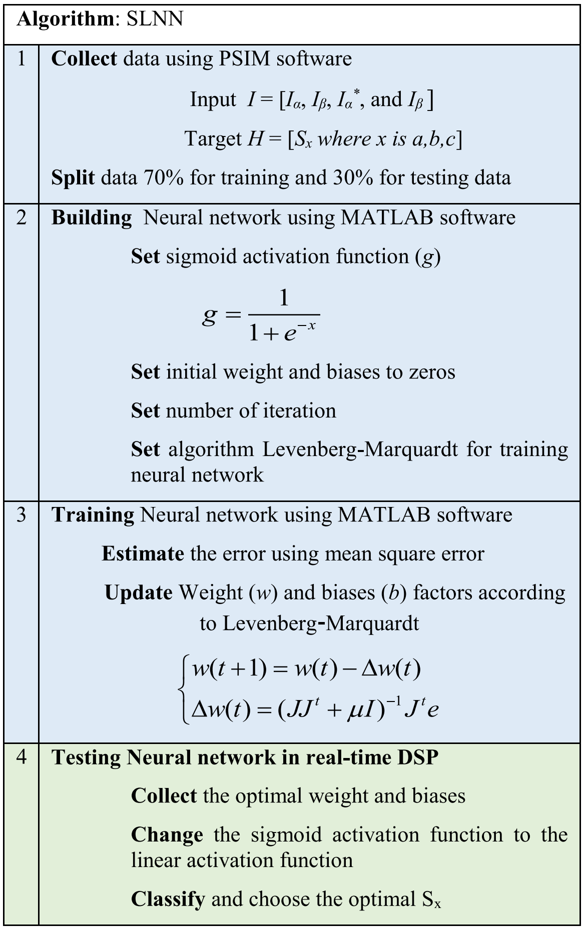
where J is the Jacobian matrix, e is the error function, $W(t+1)$ is the weight update for each iteration, μ is the combination coefficient, and I is the identity matrix.

The calculation numbers of SLNN are compared with those of MLNN as proposed in [27], and are shown in (15) and (16),

$$N_{SLNN} = HI, \tag{15}$$

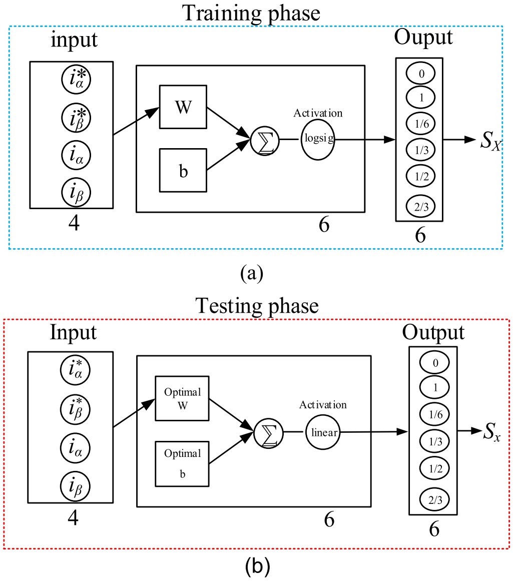
$$N_{MLNN} = 2lm + 2mH + am + bH, \tag{16}$$

where m is the hidden layer, and a and b are the number of calculations in the activation function.



```
Algorithm: SLNN
1  Collect data using PSIM software
       Input $I = [I_\alpha, I_\beta, I_\alpha^*, \text{and } I_\beta]$
       Target $H = [S_x \text{ where } x \text{ is } a,b,c]$
   Split data 70% for training and 30% for testing data
2  Building Neural network using MATLAB software
       Set sigmoid activation function ($g$)
           $g = \frac{1}{1+e^{-x}}$
       Set initial weight and biases to zeros
       Set number of iteration
       Set algorithm Levenberg-Marquardt for training neural network
3  Training Neural network using MATLAB software
       Estimate the error using mean square error
       Update Weight ($w$) and biases ($b$) factors according
           to Levenberg-Marquardt
           $w(t+1) = w(t) - \Delta w(t)$
           $\Delta w(t) = (JJ^t + \mu I)^{-1} J^t e$
4  Testing Neural network in real-time DSP
       Collect the optimal weight and biases
       Change the sigmoid activation function to the linear activation function
       Classify and choose the optimal $S_x$
```

FIGURE 6. Process of SLNN.



(a)

(b)

FIGURE 7. SLNN structure: (a) Training phase; and (b) Testing phase.

D. TRAINING PROCESS OF SLNN

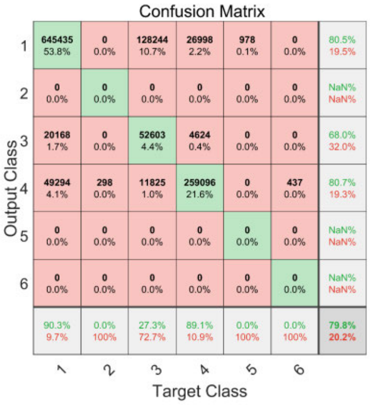
The SLNN was iteratively trained offline using MATLAB software. The input parameters are I_α, I_β, I_α^*, and I_β^* currents

TABLE 1. Dataset distribution.

Dataset	Frequency	Percent
Training	840000	70%
Testing	360000	30%

TABLE 2. Simulation specifications.

Parameters	Value
Rated Power	11 [kw]
Pole (P)	6
d-axis inductance (L_d)	0.01317 [H]
q-axis inductance (L_q)	0.0156 [H]
Stator resistance (R_s)	0.349 [Ω]
Magnetic flux linkage (φ_f)	0.554 [Wb]
Based speed	875 [rpm]
Sampling time (T_s)	100 [μs]
DC-link voltage (V_{dc})	300 [V]



	1	2	3	4	5	6	
1	645435 53.8%	0 0.0%	128244 10.7%	26998 2.2%	978 0.1%	0 0.0%	80.5% 19.5%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	20168 1.7%	0 0.0%	52603 4.4%	4624 0.4%	0 0.0%	0 0.0%	68.0% 32.0%
4	49294 4.1%	298 0.0%	11825 1.0%	259096 21.6%	0 0.0%	437 0.0%	80.7% 19.3%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	90.3% 9.7%	0.0% 100%	27.3% 72.7%	89.1% 10.9%	0.0% 100%	0.0% 100%	79.8% 20.2%

FIGURE 8. Confusion matrix of the proposed SLNN-MPC-DSVPWM for 37 VVs.

in a stationary reference frame, where the target is the optimal value of the switching signals S_a, S_b, and S_c. The parameter settings for SLNN contained 10 iterations. Increasing the number of iterations to a value greater than 10 did not lead to any further improvement. Therefore, 10 was selected as the optimal value. The data were split into 70% for training and 30% for testing as shown in Table 1. In addition, it is important to test the performance using a PSIM simulation with different current and speed values to ensure that the SLNN is learning and not memorizing. Furthermore, in the training process, the training set should be large (e.g., tens or hundreds of thousands or more), uniform, and cover different current and speed values. Therefore, More than one million data points were collected under different conditions. An adequate training process prevents the SLNN from overfitting and underfitting and improves the stability of the proposed method. The SLNN-MPC showed good accuracy in the overall performance for switching state a, with an accuracy of approximately 80% as shown in Fig. 8. Switching state b and c have also similar accuracy. In general, MPC–DSVPWM requires observers to compensate for parameter variations in motor-drive applications. SLNN-MPC-DSVPWM can address this issue by adding new input elements to correct the system parameters using observers [27].

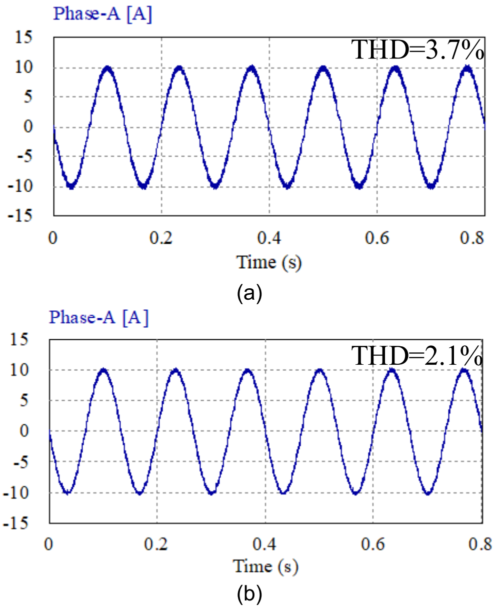


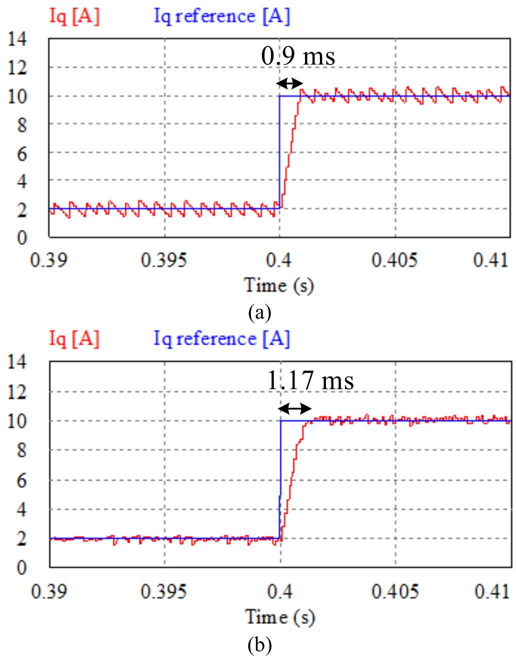
FIGURE 9. Simulation results of *A*-phase current waveform: (a) MPC-DSVPWM; and (b) SLNN-MPC-DSVPWM.

IV. SIMULATION RESULTS

The effectiveness of the proposed SLNN-MPC-DSVPWM was analyzed and proved by simulation results obtained using the PSIM tool. The performance of the proposed method was compared with that of the MPC-DSVPWM. The simulation parameters are listed in Table 2.

Table 3 lists the performance of each activation function in the testing phase, showing total harmonic distortion for the$A$-phase current, and the rise time of the$I_q$current. For 37 VVs,Linear activation function exhibited a THD of 2.1% similar as fast sigmoid and sigmoid activation functions with THD of 2% and 2.1%, respectively. However, linear activation function has faster dynamic response of 1.17 ms compared to fast sigmoid and sigmoid activation function with dynamic response of 1.2 ms and 1.6 ms, respectively.The linear activation function provides better performance than the sigmoid and fast sigmoid activation functions in the testing phase. However, in the training phase, the sigmoid function was used to achieve good learning between the input and output of the MPC-DSVPWM. The main use of linear activation function in the testing phase is to reduce the execution time of the proposed model without affect the performance of the model which clearly proved in Table 3.

Figs. 9 and 10 show the A-phase and I $_q$ current waveforms, respectively. Fig. 9 shows I_q performance during the steady-state response of the MPC-DSVPWM and the proposed SLNN-MPC-DSVPWM, respectively. The speed was 150 rpm and the I_q^* was 10 A. SLNN-MPC-DSVPWM



(a)

(b)

FIGURE 10. Simulation results of I_q current waveform: (a) MPC-DSVPWM; and (b) SLNN-MPC-DSVPWM.

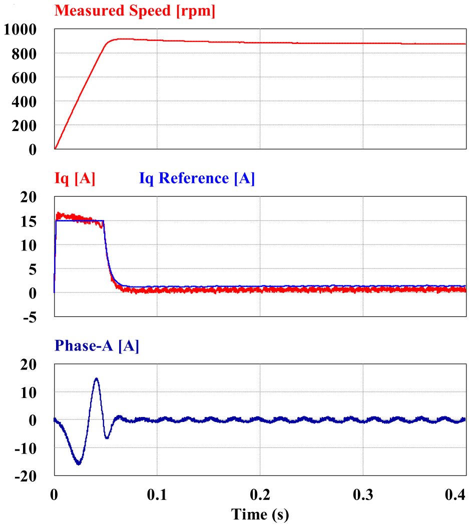


FIGURE 11. Simulation results for speed variation from 0 to 875 rpm under no load for proposed SLNN-MPC-DSVPWM with 37 VVs.

TABLE 3. Comparison results for different activation functions.

Activation function	Number of VVs	I_A THD	I_q Rise time
Sigmoid	12	5.6%	0.9 ms
Fast sigmoid		5.6%	0.9 ms
linear		5.6%	0.9 ms
Sigmoid	37	2.1%	1.6 ms
Fast sigmoid		2%	1.2 ms
linear		2.1%	1.17 ms

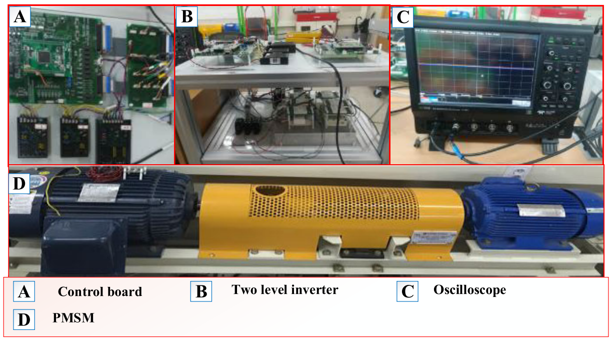


FIGURE 12. Experimental setup.

exhibited better performance than MPC-DSVPWM with THD of 2.1% compared with 3.7%. A percentage of reduction in the THD of 43% for the proposed model using 37 VVs compared with conventional MPC-DSVPWM using 12 VVs

For the dynamic response performance, I_q^* was increased from 2 A to 10 A at 150 rpm as shown in Fig. 10. Both methods demonstrated fast dynamic response for I_q current waveform. MPC-DSVPWM has faster response of 0.9 ms compared to 1.17 ms for SLNN-MPC-DSVPWM. In addition, a clear reduction in the I_q current ripple was observed for the proposed SLNN-MPC-DSVPWM compared with MPC-DSVPWM. The performance of SLNN-MPC-DSVPWM with varying motor speeds is shown in Fig. 11. The speed varies from 0 to 875 rpm (i.e., the base speed) in accordance with the applied DC-link voltage. From results, it can be seen that the I_q current waveform follows the I_q^* current. In addition, a small I_q current ripple with a sinusoidal A-phase current was achieved using SLNN-MPC-DSVPWM. As a result, the 80% accuracy of the SLNN has demonstrated a better performance in terms of THD reduction for the MPC-DSVPWM when large VVs are used. In addition, the proposed method successfully implemented under various operational conditions, as verified by the simulation results.

V. EXPERIMENTAL RESULTS

The advantages of the proposed SLNN-MPC-DSVPWM over the conventional MPC-DSVPWM and recent MLNN-MPC were verified experimentally. Fig. 12 shows the experimental setup using a two-level inverter and a low-order DSP TMS320F28335. To load the machine, an induction motor (IM) controlled by a commercially available YASUKAWA inverter was used. Due to hardware limitations, the applied DC-link voltage was 300 V, hence the base speed was reduced to 875 rpm. In addition, because of the high execution time of MPC-DSVPWM and MLNN-MPC-DSVPWM, only up to 12 VVs can be implemented in this DSP module. Meanwhile, the execution time was calculated using MPC-DSVPWM without running the full system where the algorithm was applied to the DSP to calculate the execution time at different VVs. The proposed SLNN-MPC-DSVPWM with 37 VVs was smoothly applied and operated due to its low execution time.

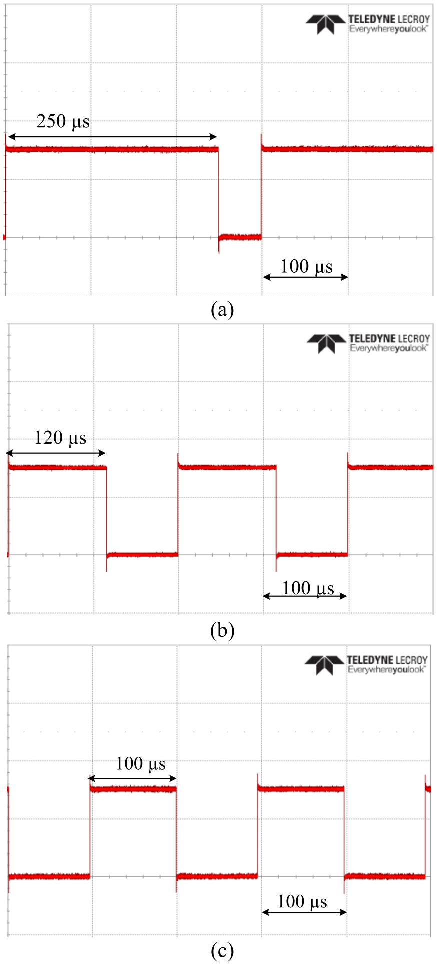


FIGURE 13. Experimental results of execution time for recent MLNN-MPC for different MLNN structures: (a) 8/42/6 [27]; (b) 8/15/7 [26]; and (c) 4/15/7 [25].

A. EXECUTION TIME OF RECENT MLNN-MPC STRUCTURES

Fig. 13 shows the execution time for recent MLNN-MPC implemented in TMS320F28335. The execution time with a MLNN structure of 8/42/6 is 250 μs, where 8 is the input followed by 42 as the number of neurons in the hidden layer and 6 as the output. The tanh activation function is used in the hidden layer and the sigmoid activation function is used in the output layer [27]. It is clearly observed that the execution time is heavily affected by the number of neurons in the hidden layers as shown in Fig. 13a [27]. In [27] used 42 neurons in the hidden layer which requires large matrix multiplication and calculation of sigmoid function which needs large execution time especially when low order DSP and serial execution are used. Parallel execution time have to be adopted in order to have an execution time less than 100 μs. In addition, the other two MLNN structures of 4/15/7 and 8/15/7, presented in the literature [25], [26]. Their execution time is obtained as shown in Fig. 13b and Fig. 13c, respectively. From these results, it is clearly shown that MLNN-MPC cannot be implemented in low-order DSPs if a large number of neurons in the hidden layer are used. In addition, using a larger number of VVs affects significantly the execution time of MLNN due to the number of VVs is equal to the number of outputs as in [25] and [26]. However, the output of the proposed method is the switching states instead of VVs which reduces the execution time significantly.

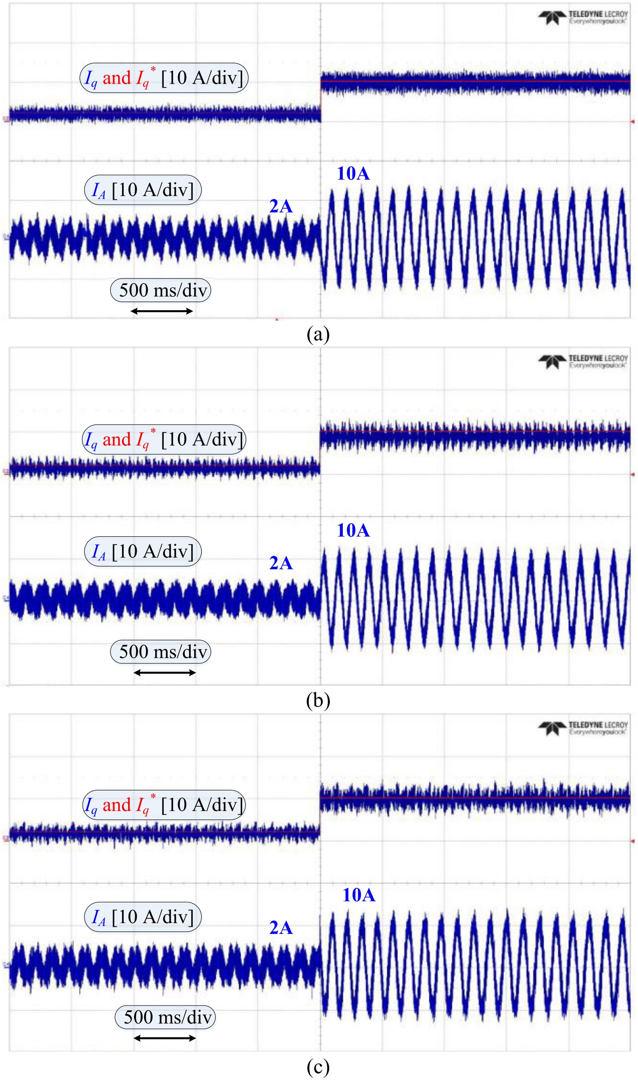


FIGURE 14. Experimental results of A-phase and I_q current waveform for seven VVs: (a) Conventional MPC (b) MLNN-MPC; and (c) SLNN-MPC.

B. PERFORMANCE COMPARISON FOR A SMALL NUMBER OF VVs

A comparison between MPC, MLNN-MPC, and SLNN-MPC for a small number of VVs is implemented first in order to show the superiority of SLNN-MPC compared with MLNN-MPC for small number of VVs. Seven VVs are used and the I_q^* was increased from 2 A to 10 A at 150 rpm. MLNN structure that is used is 4/2/7, where 4 is the input followed by 2 as the number of neurons in the hidden layer and 7 as the output. The tanh activation function is used in the hidden layer and the sigmoid activation function is used in the output layer [27].

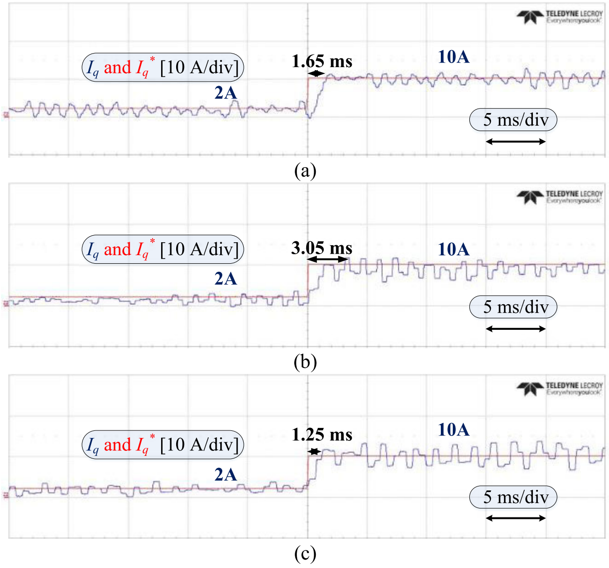


FIGURE 15. Experimental result of I_q current waveform in transient response for seven VVs: (a) Conventional MPC, (b) MLNN-MPC, and (c) SLNN-MPC.

Fig. 14 shows the experimental results of I_q and A-phase current waveforms during the dynamic response for each method. The MLNN-MPC and SLNN-MPC have relatively similar steady-state performance with MPC. However, I_q current waveform of MLNN-MPC does not follow I_q^* perfectly as compared with SLNN-MPC. Fig. 15 shows the magnified experimental results for the transient response of I_q from 2 to 10A. It is clear evident that SLNN-MPC and MLNN-MPC can learn the fast transient response of the MPC. However, SLNN-MPC has the fastest transient response with 1.25 ms compared to MLNN-MPC and MPC-DSVPWM with 3.05 ms and 1.65 ms, respectively. Slow dynamic performance of MLNN-MPC is achieved because the I_q current waveform does not track the I_q^* current waveform at the center.

C. PERFORMANCE COMPARISON FOR A LARGE NUMBER OF VVs

Importantly, due to larger execution time of MLNN-MPC-DSVPWM and MPC-DSVPWM for large number of VVs, only proposed SLNN-MPC-DSVPWM can be used for 37 VVs. Fig. 16 and Fig. 17 show the experimental results of I_q, A-phase current waveform, THD, and FFT, for a large number of VVs respectively. SLNN-MPC-DSVPWM showed a better performance than MPC-DSVPWM with THD of 10% compared with 28.9%, respectively. In addition, the frequency spectrum located within the first 10 kHz division shows a noticeable reduction of the harmonics of the proposed SLNN-MPC-DSVPWM compared with MPC-DSVPWM. However, MPC-DSVPWM has faster response time compared with SLNN-MPC-DSVPWM. Furthermore, the proposed SLNN-MPC-DSVPWM was tested with speed variation, as shown in Fig. 18. It can be observed that the I_q current waveform perfectly follows its desired value I_q^*.

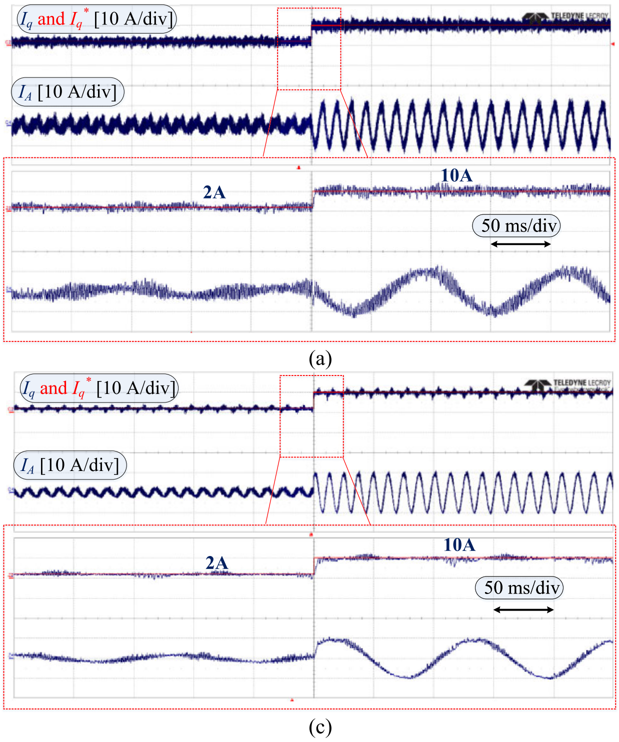


FIGURE 16. Experimental results of A-phase and I_q current waveform for: (a) MPC-DSVPWM with VVs=12; (b) SLNN-MPC-DSVPWM with VVs=37.

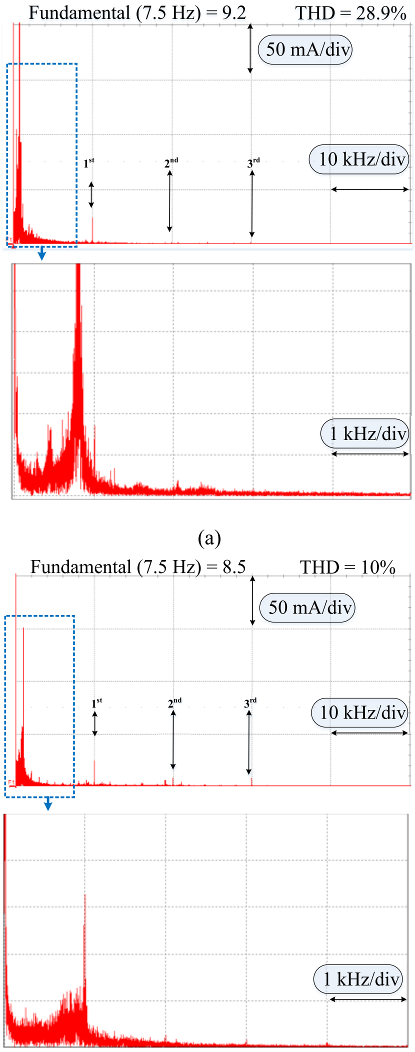
Fig. 19 shows the execution times for each method with different numbers of VVs. The execution time of SLNN-MPC-DSVPWM for 12VV is only 7 μs compared with 30 μs for MPC-DSVPWM. In addition, A maximum of 8 μs was required for all the calculations of the proposed method for 37 VVs compared with 75 μs using conventional MPC-DSVPWM with the same number of VVs. The reason for the low execution time of SLNN-MPC-DSVPWM is that it does not require lookup table initialization compared with MPC-DSVPWM. In addition, the linear activation function significantly reduces the execution time compared with the sigmoid and fast sigmoid activation functions. Moreover, SLNN-MPC-DSVPWM can implement higher step prediction horizons to improve system performance without increasing the calculation burden. However, for MPC-DSVPWM method, the implementation of higher step prediction horizons increases the number of possible switches exponentially which increases the computation time exponentially.

D. AFFECT OF ACTIVATION FUNCTION ON THE EXECUTION TIME

The activation function significantly affects the size of the execution time. As shown in Table 4, the linear activation function took approximately 8 μs compared with 80 μs and 40 μs for the sigmoid and fast sigmoid activation functions, respectively. However, in the training phase, the sigmoid function was used to achieve good learning between the input and output of the MPC-DSVPWM.

TABLE 4. Execution time for different activation functions.

Activation function	Number of VVs	Execution time
Sigmoid	12	30 µs
Fast sigmoid		20 µs
linear		7 µs
Sigmoid	37	80 µs
Fast sigmoid		40 µs
linear		8 µs



(a)

(b)

FIGURE 17. Experimental results of THD for: (a) MPC-DSVPWM with 12VVs (b) SLNN-MPC-DSVPWM with 37VVs.

E. AFFECT OF HIDDEN NEURONS AND OUTPUT LAYER ON THE EXECUTION TIME

The hidden neurons significantly affects the performance of the execution time. As shown in Fig. 13, when the number of hidden neurons increase from 15 to 42 neurons, the execution time was increase from the 120 μs to 200 μs. Therefore, the MLNN structure proposed in previous studies could not be implemented for low sampling time when low-order DSP and serial execution time are used. Moreover, if MLNN-MPC used for larger number of VVs, the execution time of MLNN increases significantly due to the number of VVs are equal to the number of outputs as in [25] and [26].

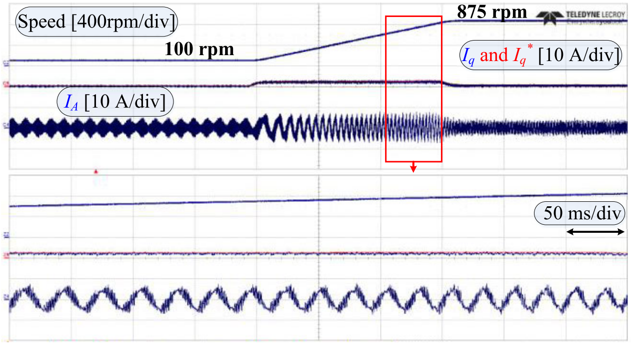
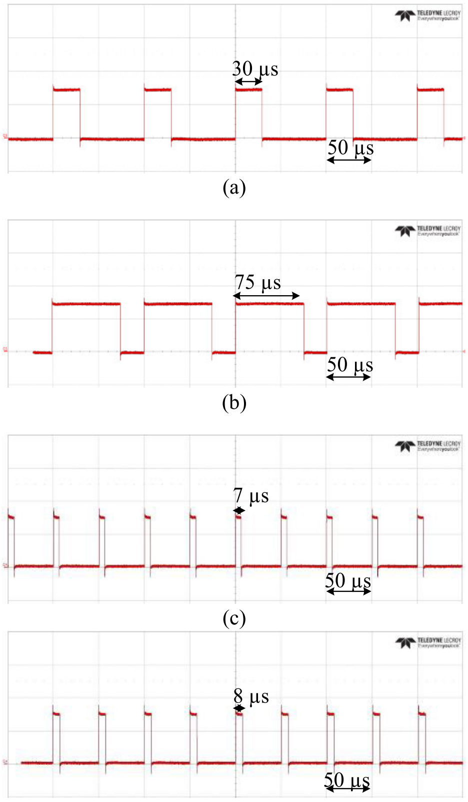


FIGURE 18. Experimental results for speed variation from 100 to 875 rpm under no load for SLNN-MPC-DSVPWM.



(a)

(b)

(c)

(d)

FIGURE 19. Experimental results of execution time for: (a) MPC- DSVPWM with VVs=12; (b) MPC-DSVPWM with VVs=37; (c) SLNN-MPC-DSVPWM with VVs=12; and (d) SLNN-MPC-DSVPWM with VVs=37.

Consequently, The MLNN structure could not be implement if 37 VVs are used. The proposed method solves this limitation by using three SLNN to avoid lookup table initialization for a large number of virtual VVs. In addition, the proposed method does not require a large multiplication matrix of the input with all VVs. Instead, it simply multiplies the input with suitable switch states for each phase.

F. AFFECT OF NUMBER OF VVs ON THE EXECUTION TIME

The main limitation of MPC-DSVPWM is the execution time increases exponentially when a large number of VVs are used. In contrast, A maximum of 8 μs was only required for all the calculations of the proposed method for 37 VV compared with 75 μs using conventional MPC-DSVPWM with the same number of VVs.

VI. CONCLUSION

A creative control method for the current control of a PMSM was proposed in this study. The proposed SLNN-MPC-DSVPWM can be implemented using a low-order microprocessor for a large number of VVs. The proposed method used SLNNs to avoid lookup table initialization of all VVs and large matrix multiplication. In addition, the use of the linear activation function in the testing phase while using the sigmoid activation in the training phase significantly reduces the execution time compared with other previous research studies which use high computation activation functions in both the training and testing phases. The effectiveness of the proposed method was verified through simulations and experimental results. The proposed SLNN-MPC-DSVPWM shows a promising steady-state performance with an execution time of only 8 μs. The limitation of the proposed method is the training process based on MPC-DSVPWM which in general requires observers to compensate for parameter variations in motor-drive applications. Therefore, adaptive neural networks should be used in a future direction. Adaptive SLNN-MPC-DSVPWM can add new input elements to correct the system parameters using observers. In addition, a SLNN-MPC-DSVPWM can implement higher step prediction horizons to improve system performance without increasing the calculation burden which can be as a future improvement of the method.

REFERENCES

[1] S. S. Hakami and K.-B. Lee, "Four-level hysteresis-based DTC for torque capability improvement of IPMSM fed by three-level NPC inverter," *Electronics*, vol. 9, no. 10, p. 1558, Sep. 2020.

[2] D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: Two viable schemes for induction motors torque control," *IEEE Trans. Power Electron.*, vol. 17, no. 5, pp. 779–787, Sep. 2002.

[3] Y. Inoue, S. Morimoto, and M. Sanada, "Comparative study of PMSM drive systems based on current control and direct torque control in flux-weakening control region," *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 2382–2389, Nov. 2012.

[4] J. Rodriguez, C. Garcia, A. Mora, F. Flores-Bahamonde, P. Acuna, M. Novak, Y. Zhang, L. Tarisciotti, S. A. Davari, Z. Zhang, and F. Wang, "Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies," *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 3927–3942, Apr. 2022.

[5] M. Preindl and S. Bolongnani, "Model predictive direct speed control with finite control set of PMSM drive systems," *IEEE Trans. Power Electron.*, vol. 28, no. 2, pp. 1007–1015, Feb. 2013.

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

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

[8] Y. Zhang, T. Jiang, and J. Jiao, "Model-free predictive current control of DFIG based on an extended state observer under unbalanced and distorted grid," *IEEE Trans. Power Electron.*, vol. 35, no. 8, pp. 8130–8139, Aug. 2020.

[9] P. Karamanakos and T. Geyer, "Model predictive torque and flux control minimizing current distortions," *IEEE Trans. Power Electron.*, vol. 34, no. 3, pp. 2007–2012, Mar. 2019.

[10] F. Wang, H. Xie, Q. Chen, S. A. Davari, J. Rodriguez, and R. Kennel, "Parallel predictive torque control for induction machines without weighting factors," *IEEE Trans. Power Electron.*, vol. 35, no. 2, pp. 1779–1788, Feb. 2020.

[11] Y. Zhang, B. Zhang, H. Yang, M. Norambuena, and J. Rodriguez, "Generalized sequential model predictive control of IM drives with field-weakening ability," *IEEE Trans. Power Electron.*, vol. 34, no. 9, pp. 8944–8955, Sep. 2019.

[12] Y. Zhang, Y. Bai, and H. Yang, "A universal multiple-vector-based model predictive control of induction motor drives," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 6957–6969, Aug. 2018.

[13] Y. Zhang and H. Yang, "Model predictive torque control of induction motor drives with optimal duty cycle control," *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6593–6603, Dec. 2014.

[14] Y. Zhang and H. Yang, "Two-vector-based model predictive torque control without weighting factors for induction motor drives," *IEEE Trans. Power Electron.*, vol. 31, no. 2, pp. 1381–1390, Feb. 2016.

[15] Y. Zhang and H. Yang, "Generalized two-vector-based model-predictive torque control of induction motor drives," *IEEE Trans. Power Electron.*, vol. 30, no. 7, pp. 3818–3829, Jul. 2015.

[16] I. Osman, D. Xiao, K. S. Alam, M. P. Akter, S. M. S. I. Shakib, and M. F. Rahman, "Discrete space vector modulation based model predictive torque control with no sub-optimization," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8164–8174, Oct. 2020.

[17] S. Vazquez, J. I. Leon, L. G. Franquelo, J. M. Carrasco, O. Martinez, J. Rodriguez, P. Cortes, and S. Kouro, "Model predictive control with constant switching frequency using a discrete space vector modulation with virtual state vectors," in *Proc. IEEE Int. Conf. Ind. Technol.*, Feb. 2009, pp. 1–6.

[18] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Nov. 2017.

[19] I. Osman, D. Xiao, M. F. Rahman, M. Norambuena, and J. Rodriguez, "Discrete space vector modulation based model predictive flux control with reduced switching frequency for IM drive," *IEEE Trans. Energy Convers.*, vol. 36, no. 2, pp. 1357–1367, Jun. 2021.

[20] Y. Wang, X. Wang, W. Xie, F. Wang, M. Dou, R. M. Kennel, R. D. Lorenz, and D. Gerling, "Deadbeat model-predictive torque control with discrete space-vector modulation for PMSM drives," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3537–3547, May 2017.

[21] H.-C. Moon, J.-S. Lee, and K.-B. Lee, "A robust deadbeat finite set model predictive current control based on discrete space vector modulation for a grid-connected voltage source inverter," *IEEE Trans. Energy Convers.*, vol. 33, no. 4, pp. 1719–1728, Dec. 2018.

[22] I. M. Alsofyani and K.-B. Lee, "A unidirectional voltage vector preselection strategy for optimizing model predictive torque control with discrete space vector modulation of IPMSM," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 12305–12315, Dec. 2021.

[23] 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, Dec. 2021.

[24] T. Dragičević and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Trans. Ind. Electron*, vol. 66, no. 11, pp. 2124–2142, Oct. 2018.

[25] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, 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.

[26] 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.

[27] D. Wang, Z. J. Shen, X. Yin, S. Tang, X. Liu, C. Zhang, J. Wang, J. Rodriguez, and M. Norambuena, "Model predictive control using artificial neural network for power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3689–3699, Apr. 2022.

[28] L. M. Halabi, I. M. Alsofyani, and K.-B. Lee, "Hardware implementation for hybrid active NPC converters using FPGA-based dual pulse width modulation," *J. Power Electron.*, vol. 21, no. 11, pp. 1669–1679, Nov. 2021.

[29] A. J. Stothers, "On the complexity of matrix multiplication," Ph.D. dissertation, School Math., Univ. Edinburgh, Edinburgh, Scotland, 2010.

[30] H. A. G. Al-kaf, K. S. Chia, and N. A. M. Alduais, "A comparison between single layer and multilayer artificial neural networks in predicting diesel fuel properties using near infrared spectrum," *Petroleum Sci. Technol.*, vol. 36, no. 6, pp. 411–418, Mar. 2018.

[31] S.-H. Kim, *Electric Motor Control: DC, AC, and BLDC Motors*. Amsterdam, The Netherlands: Elsevier, 2017.

HASAN ALI GAMAL AL-KAF (Graduate Student Member, IEEE) received the B.Eng. and master's degrees in electronic engineering from the Universiti Tun Hussein Onn Malaysia, Malaysia. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Ajou University, Suwon, South Korea. His research interests include power electronics, artificial intelligence, and machine learning.

KYO-BEUM LEE (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical and electronic engineering from Ajou University, Suwon, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in electrical engineering from Korea University, Seoul, South Korea, in 2003. From 2003 to 2006, he was with the Institute of Energy Technology, Aalborg University, Aalborg, Denmark. From 2006 to 2007, he was with the Division of Electronics and Information Engineering, Jeonbuk National University, Jeonju, South Korea. In 2007, he joined the Department of Electrical and Computer Engineering, Ajou University. His research interests include electric machine drives, renewable power generations, and electric vehicle applications. He is the Editor-in-Chief of the *Journal of Power Electronics*. He is an Associate Editor of the IEEE TRANSACTIONS ON POWER ELECTRONICS.