Artificial Neural Network Based FCS-MPC for 3-Level Inverters

Xinliang Yang[1] · Kun Wang[1] · Jongseok Kim[1] · Ki-Bum Park[1*]

Abstract

Finite control set model predictive control (FCS-MPC) stands out for fast dynamics and easy inclusion of multiple nonlinear control objectives. But for long horizontal prediction or complex topologies with multi-levels and multi-phases, the required computation burden surges exponentially as the increases of candidate switch states during one control period. This leads longer sample period to guarantee enough time for traverse progress of cost function minimization. In other words, the allowed highest switching frequency is bounded much far from the physical limits, especially for wide-band semiconductors applications. To overcome this, the parallel computing characteristic of artificial neural network (ANN) motivates the idea of ANN-based FCS-MPC imitator (ANN-MPC). In this article, by the utilization of a shallow neural network, we implement ANN-MPC on a neutral point clamped (NPC) converter. The expert (FCS-MPC) is designed at first, and then the basic structure including activation function selection, training data generation and offline training progress, and online operation of the imitator (ANN-MPC) are discussed. After the design of both expert and imitator, a comparative analysis is conducted by FPGA-in-the-loop implementation in MATLAB/Simulink environment. The verification results of ANN-MPC show highly similarly qualified control performance and much reduced computation resources requirement.

Keywords Model predictive control · Power converter · Artificial neural network · Computation burden · Imitation.

1 Introduction

The finite control set model predictive control (FCS-MPC) is becoming more and more promising with the development of microprocessors. Among a number of advanced control strategies, it stands out because of the strong ability of multiple nonlinear control objectives inclusion and outstanding transient state control performance [1-4]. By the minimization of a weighted cost function, FCS-MPC traverses all potential switch states and selects the optimal one for the next whole control period.

However, the practical applications of FCS-MPC remain stagnant within relatively simple topological converters and short horizontal prediction. For long horizontal prediction, the computation burden surges as the increase of predictive step amounts. While facing multi-level or multi-phase power converters, the number of candidate switch states would increase exponentially, which requires much high computation demands for the traversal progress of cost function minimization. And there would be trade-off progress between the complexity of converters and feasible highest switching frequency, in another word, the generality and the power density, which is against the trend of wide-band semiconductor-based power system design in this decade.

Recently, the artificial neural network (ANN) has attracted much attention in power electronics applications [5, 6]. According to the universal approximation theorem [7], even a shallow neural network (with only one hidden layer) could achieve a precisely approximation for mapping of complex input-output. And its parallel propagation (parallel computation) characteristic motivates renovation of traditional controller implementation, automated design of system and circuit components, and so on. For example, the ANN-based surrogate model is used to replace lookup tables for inductor thermal and magnetic models [8-10]. And system parameter identification could be achieved without prior knowledge by the assistance of ANN [11].

If the FCS-MPC is regarded as an expert to generate training data, an ANN-based imitator of FCS-MPC (ANN-MPC) had been discussed for resonant converters [12], simple two level three-phase inverters [13, 14], multicell converters [15], modular multilevel converters [16] and multi-step prediction [17]. The control performance of ANN-MPC in other popular topologies and their variants needs to be investigated.

In this work, we closely investigate the performance of ANN-MPC for a three-phase three-level neutral point

* Ki-Bum Park
ki-bum.park@kaist.ac.kr

1 CCS Graduate School of Mobility, KAIST, Daejeon, South Korea

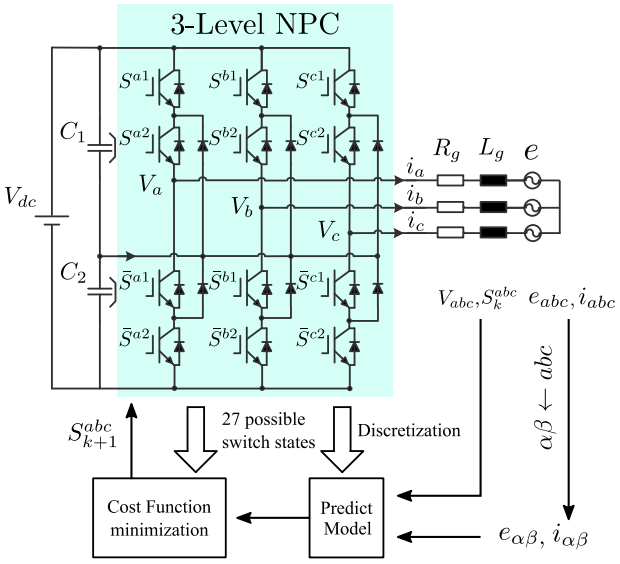

Fig. 1 Overview of FCS-MPC for a 3-L NPC converter connected to resistive–inductive–active loads

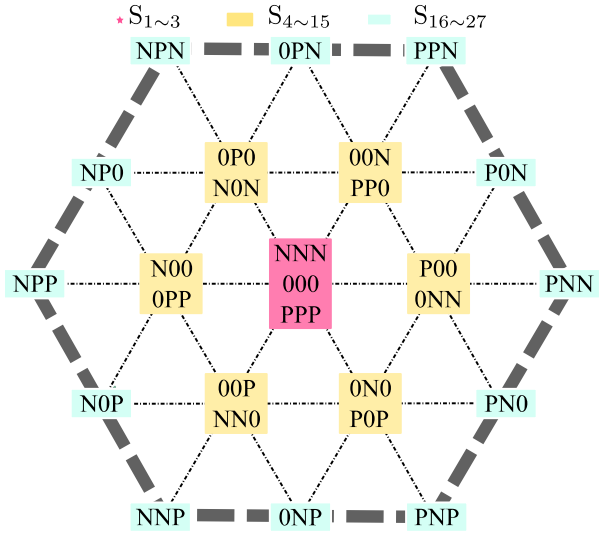

Fig. 2 Definition of 27 switching states and corresponding voltage vectors

clamped (NPC) power converters. A suitable FCS-MPC controller is designed for the NPC converter and corresponding load environment, which supplies training data for ANN. Then ANN-MPC and FCS-MPC are parallel simulated to collect new training sets, especially for mismatched control outputs for continuous reciprocating training. The FPGA-in-the-loop (FIL) simulation results show that the trained neural network accurately imitates FCS-MPC's input-output mapping while achieving qualified control performance and much reducing the computation burden.

2 CONVENTIONAL FCS-MPC CONTROLLER DESIGN

According to if the control input is continuous, MPC could be divided into continuous control sets (CCS-MPC) and FCS-MPC. In this article, the latter one is chosen as an expert imitated by a shallow neural network. And the core design steps of FCS-MPC are discussed in this section from discrete model construction to weighting factors design.

Table 1 System Parameters

Parameter	Value
DC-link voltage	$V_{dc} = 800$ V
DC-link capacitance	$C_1 = C_2 = 3.1\ mF$
Load passive parameter	$R_g = 0.1\ \Omega, L_g = 8$ mH
EMF Voltage	$f_{ref} = 50$Hz, $\hat{E}_{line} = 380$V
Sampling time	$T_s = 20\ \mu$s
Weighting factor	$\lambda_{dc} = 0.1$

2.1 Discrete Converter Model

As shown in Fig.1, the 3-L NPC converter connected to resistive–inductive–active loads is chosen to show ANN-MPC's control performance. With this model, a wide range of applications could be covered including induction machine drive, passive loads, grid-connected converters, and so on. The upper and lower switches are denoted, respectively, as S^{xi} and $\bar{S}^{xi}$ ($x \in \{a,b,c\}$ and $i \in \{1,2\}$). By regarding the neutral point of DC-link as a zero voltage potential point, the single-phase output voltage (v_x) could be defined as:

$$v_x := \begin{cases} \frac{1}{2}V_{dc} & \text{if}: S^{x1} = 1 \wedge S^{x2} = 1 \\ 0 & \text{if}: S^{x1} = 0 \wedge S^{x2} = 1 \\ -\frac{1}{2}V_{dc} & \text{if}: S^{x1} = 0 \wedge S^{x2} = 0 \end{cases} \quad (1)$$

where the output voltage of total 27 candidate switch states and their corresponding voltage vectors are shown in Fig. 2. All of them are divided into two categories, the redundant vector ($S_{4\sim15}$ shown in Fig.2) means vectors that show the same effect for current tracking but different for voltage balance of the DC-link.

For FCS-MPC applications, usually, the sampling frequency is relatively high [1], which enables discretization for derivative equation models from the basic electric principles like Kirchhoff laws. Here the forward Euler approximation ($\frac{d}{dt}x(t) \approx \frac{x[k+1]-x[k]}{T_s}$) is utilized for the derivative of load current. After Clarke coordinate transformation for system parameter decoupling, the result discrete system model in $\alpha\beta$ frame is as follows:

$$i_{\alpha\beta}(k+1) = \left(1 - R_g\frac{T_s}{L_g}\right) i_{\alpha\beta}(k) + \frac{T_s}{L_g}\left(v_{\alpha\beta}(k) - e_{\alpha\beta}(k)\right) \quad (2)$$

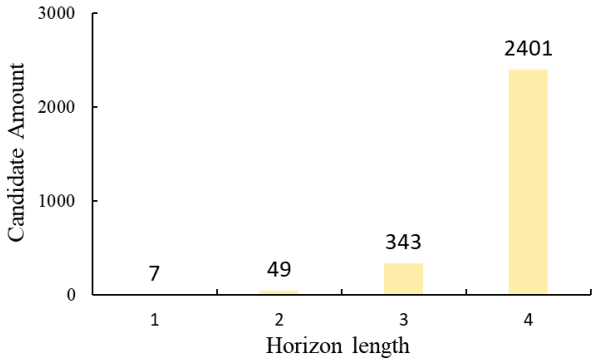

Fig. 3 Amounts of candidate switch states of conventional FCS-MPC with long prediction horizon for the three-phase 2-L inverter

$$V_o(k+1) = V_o(k) - \frac{T_s}{C}\left(i_{abc}(k)\right)^{\top}|V_{abc}(k)|, \qquad (3)$$

where C, R_g, L_g and T_s denote DC-link capacitance, load resistance, load inductance, and sampling period, respectively. V_o, $i_{\alpha\beta}$ and $e_{\alpha\beta}$ are the DC-link capacitor voltage difference, load current, and electro-motive-force (EMF) voltage, which are listed in Table 1.

2.2 FCS-MPC Design

As long as the aimed control objectives could be quantified into a cost function, with the assistance of weighting factors, all of the objectives would be greedily pursued through the trade-off progress. For NPC power converters, typical control requirements are load current reference tracking and DC-link capacitor voltage balance, which are formulated as follows:

$$g = \left(i_\alpha^* - i_\alpha^p\right)^2 + \left(i_\beta^* - i_\beta^p\right)^2 + \lambda_{dc}\left|V_o^p\right|, \qquad (4)$$

where the superscript p and $*$ represent predictive and reference value for tracking type control objectives, and λ_{dc} is the weighting factor (WF) adjusting the importance of these control objectives. According to specific applications, the reference currents have various sources, such as speed and flux controllers.

The FCS-MPC's control performance highly depends on WFs selection, while till now there are no general and systematic design methods. Some strategies could be found in pieces of literature like sequential model predictive control (SMPC) [18], reinforcement learning (RL) [19], genetic algorithm (GA) [20], and so on. In this article, the WFs design details are neglected and listed in Table 1.

3 ANN-MPC CONTROLLER DESIGN

In this section, the basic structure of the shallow neural network is introduced and a reasonable selection of activation functions is discussed for layers. Due to the finite number of candidate switch states for one control period, the main task for ANN to imitate is regarded as a multi-classification.

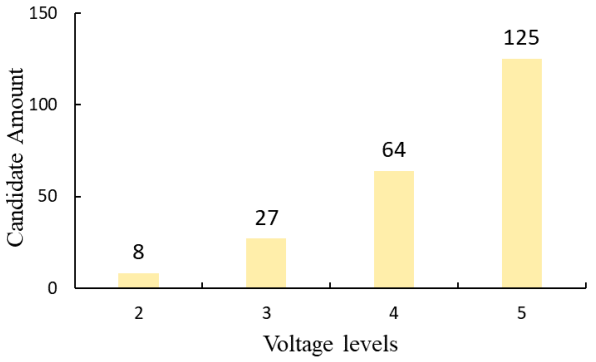

Fig. 4 Amounts of candidate switch states of conventional FCS-MPC with one-step prediction horizon for the three-phase multi-level inverters

3.1 Imitator Motivation

As long as the aimed control objectives could be quantified into a cost function, with the assistance of weighting factors, all of the objectives would be greedily pursued through the trade-off progress. For NPC power converters, typical control requirements are load current reference tracking and DC-link capacitor voltage balance, which are formulated as follows:

According to the basic control idea of FCS-MPC, all of the candidate switch states need to experience prediction and minimization stages. As Fig. 3 and Fig. 4 show, the computation burden of the traversal progress would surge exponentially with more complex topologies and longer prediction horizon, which is impossible for practical implementation. With enough number of neurons in the hidden layer of ANN, approximation ability and parallel computation characteristic enable much faster function mapping progress and high imitation accuracy at the same time. In another word, FCS-MPC is regarded as an expert, after completed training progress, ANN imitator (ANN-MPC) could theoretically show as a computationally efficient FCS-MPC, which promisingly expands FCS-MPC's generality for complex power electronics system and long horizontal prediction.

There has been a large number of various neural networks in the artificial intelligence (AI) area, such as convolutional ones and recurrent ones. However, for faster online forward propagation to beat FCS-MPC, it is unfair and not applicable to use large scale AI model, especially for simple converters and not so long prediction horizon. Therefore, here almost the simplest one, the shallow neural network, is considered to achieve the imitation task. Of course, there are some intuitive and natural idea for network structure modification, such as adding layers and adding neurons.

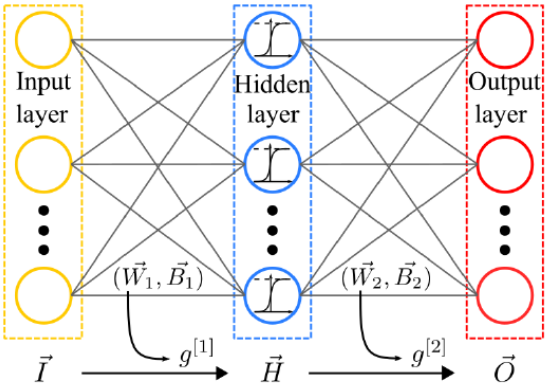

Fig. 5 Shallow neural network structure and forward propagation

Table 2 ANN Structure Parameters

Structure Component	Specification
Neuron amounts for layers	$7 \times 5 \times 27$
Hidden layer activation	Linear Pieced Sigmoid $g^{[1]}$
Output layer activation	SoftMax $g^{[2]}$

3.2 Shallow neural network structure

Usually more complex learning model means higher imitation ability, but computation burden needs to be cared, which leads us to concentrate more on lightweight learning models.

As Fig. 5 shows, a shallow neural network consists of three layers including input layer, hidden layer and output layer. The layer-to-layer propagation depends on weighted input and activation functions as follows:

$$\vec{H} = g^{[1]}\left(\langle \vec{I}, \vec{W}_1 \rangle + \vec{B}_1\right), \vec{O} = g^{[2]}\left(\langle \vec{H}, \vec{W}_2 \rangle + \vec{B}_2\right), \quad (5)$$

where $\langle \vec{x_1}, \vec{x}_2 \rangle$ represents the matrix multiplication between a vector and a matrix, $\vec{I}, \vec{H}$ and $\vec{O}$ are the feature vectors of input, output of hidden and output layers, respectively; $\vec{W}_1$ and $\vec{W}_2$ are the weight matrixes; $\vec{B}_1$ and $\vec{B}_2$ are the bias vectors; while $g^{[1]}$ and $g^{[2]}$ are the hidden and output layer's activation functions whose selection depends on appropriately matched statistic model and the input-output characteristics.

The size of input layer is determined by the information amount required by the expert controller. Here seven inputs are needed by the FCS-MPC for predictive model and cost function minimization. It is worth mentioning that, by the utilization of one-hot encoding (i.e., switch state 2 corresponds $\overbrace{[010\cdots00]}^{27\ neurons\ in\ output\ layer}$), the number of neurons in the output layer is the same as the number of switch states, which means the ANN's output is directly optimal switch state for one control period. And till now there is no systematic method to obtain the best number of neurons in the hidden layer, which we would discuss in later sections.

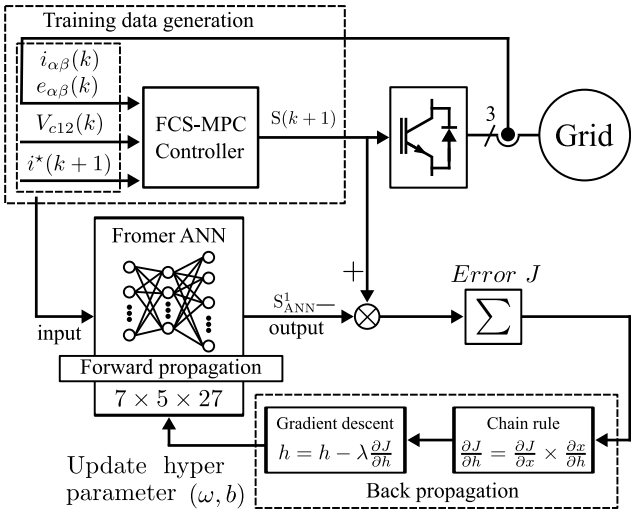

Fig. 6 Training data generation and training progress of ANN-MPC imitator

In this article, for convenient implementation on FPGA in the future, the linearly pieced sigmoid function is chosen as the activation function $g^{[1]}$ for hidden layer as follows:

$$g^{[1]}(x) = \frac{1}{1+e^{-x}} \rightarrow \vec{H} = \frac{1}{1+e^{-\left(\langle \vec{I}, \vec{W}_1 \rangle + \vec{B}_1\right)}}, \quad (6)$$

otherwise the original sigmoid function has to be stored as a large lookup table in inner FPGA, which occupies large part of storage resources. For output layer, during one control period, FCS-MPC traverses all potential candidate switch states and select one, which is more closely to be a multi-classification task motivating SoftMax function as the activation function as follows:

$$g^{[2]}(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \rightarrow (\vec{O})_i = \frac{e^{\left(\langle \vec{H}, \vec{W}_2 \rangle + \vec{B}_2\right)_i}}{\sum_{j=1}^{k} e^{\left(\langle \vec{H}, \vec{W}_2 \rangle + \vec{B}_2\right)_j}}, \quad (7)$$

where $(\vec{O})_i$ means ith component of the vector $\vec{O}$, which is similar for $\left(\langle \vec{H}, \vec{W}_2 \rangle + \vec{B}_2\right)_i$. By the utilization of SoftMax function, the output $\vec{O}$ could be regarded as a normalized probability distribution in which the element corresponding to the maximum probability is 1, the others are 0.

3.3 Training Data Generation and Training Progress

As shown in Fig. 6, through pure virtual simulation or hardware-in-the-loop simulation, the data sets for offline training of ANN are produced by the FCS-MPC expert, which has been designed in Section 2. It deserves mentioning that the designed environment, load state including electric characteristic and etc., should try to cover the whole operation range, such as no load, full load,

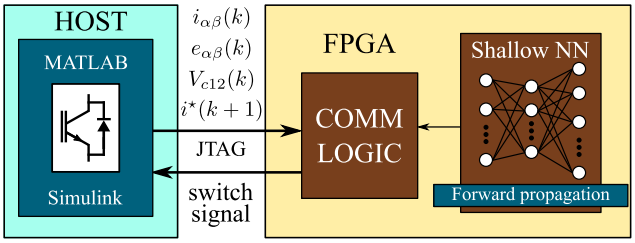

Fig. 7 System-level view of FPGA-in-the-loop (FIL) simulation environment

reference step transient, even fault situation and etc.

After wide range of training data generation, the offline training progress of a shallow neural network is continued for collection of weight matrix $\vec{W}_1$ and $\vec{W}_2$ and bias vectors $\vec{B}_1$ and $\vec{B}_2$. After forward propagation of training sets, by the utilization of back propagation combining chain rule and gradient descent, the derivative of error over hyperparameters (weight matrix and bias vectors) shows effective modification for hyperparameters. And usually for fast and stable convergence, the learning rate (modification step for hyperparameters) would be gradually reduced as the iteration of training progress proceeds.

To accelerate the training progress, after the first few training sessions, the immature ANN-MPC and FCS-MPC are simulated in parallel. Instead of independent operation, only FCS-MPC operates to control the NPC converter, while the control input of the immature ANN-MPC are the same as FCS-MPC's and the its control outputs are only recorded for mismatch detection. And this is for special training data collection when they have different control output especially during transient states. In another word, the mismatched control output often corresponds uncovered input domain during training.

4 Simulation Study and Verification

4.1 FPGA-in-the-loop Implementation

HDL Verifier based FPGA-in-the-loop (FIL) simulation provides the capability to use Simulink or MATLAB software for testing designs in real hardware for any existing HDL codes. The simulation set-up has been devised around a development board (DE1-SoC) with an embedded Altera Cyclone V 5CSEMA5F31C6N FPGA. This device is driven by an on-board 50 MHz oscillator and supports JTAG Mode. It has 85K programmable logic elements and 4450 Kbits embedded memory which can accommodate the function of simple controller.

As Fig. 7 shows, by the replacement of MATLAB Function block by FIL block, there would be interactive data flow between the particular FPGA and constructed Simulink environment. Here FPGA is mainly responsible for forward propagation of the designed ANN. And the

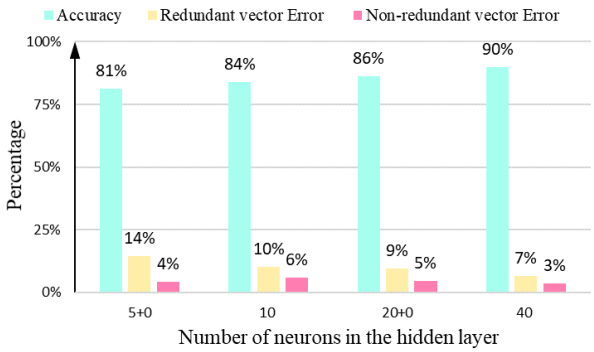

Fig. 8 ANN-MPC imitation accuracy with different neuron amounts in hidden layer

control signal (selection of switch states) is then transmitted to Simulink model by JTAG connection. The dataflow inside FPGA is monitored by an IP core generated by the HDL Verifier.

4.2 Comparative Analysis of Rough Estimation of Computation Burden

From the design of artificial neural network to the actual implementation on the controller (such as FPGA based), in this paper, we estimate the computational complexity by counting the total number of floating-point operations (FLOPs).

For traditional FCS-MPC, due to the traverse progress, each candidate switch state would experience prediction equation and cost function. As presented in [4], the computation burden ($n_{prediction}$) for former stage depends on the discretized system model, while that ($n_{cost\ function}$) of the latter one depends on complexity of cost function, such as selected order of the norm, number of control objectives and so on. Considering the number of potential switch states is $2^{y(x-1)}$, where y is the number of phases (or bridges) and x is the number of independent switch pairs for each phase, the estimated computation burden of FCS-MPC is as:

$$N_{\mathrm{FCS-MPC}} = 2^{y(x-1)}\left(n_{\mathrm{prediction}} + n_{\mathrm{cost\ function}}\right), \quad (8)$$

where $N_{\mathrm{FCS-MPC}}$ surges exponentially as the increasing number of levels or phases of power converters.

For the shallow neural network, without online learning or update, only forward propagation is needed for online implementation. The computation burden comes from matrix multiplication and activation functions as:

$$N_{\mathrm{ANN-MPC}} = 2lm + 2mn + \varepsilon_1 m + \varepsilon_2 n, \quad (9)$$

where l, m and n are the number of neurons in the input layer, hidden layer and output layer, respectively; ε_1 and ε_2 are FLOPs calculation demand of activation function $g^{[1]}$ and $g^{[2]}$ (in this paper, both ε_1 and ε_2

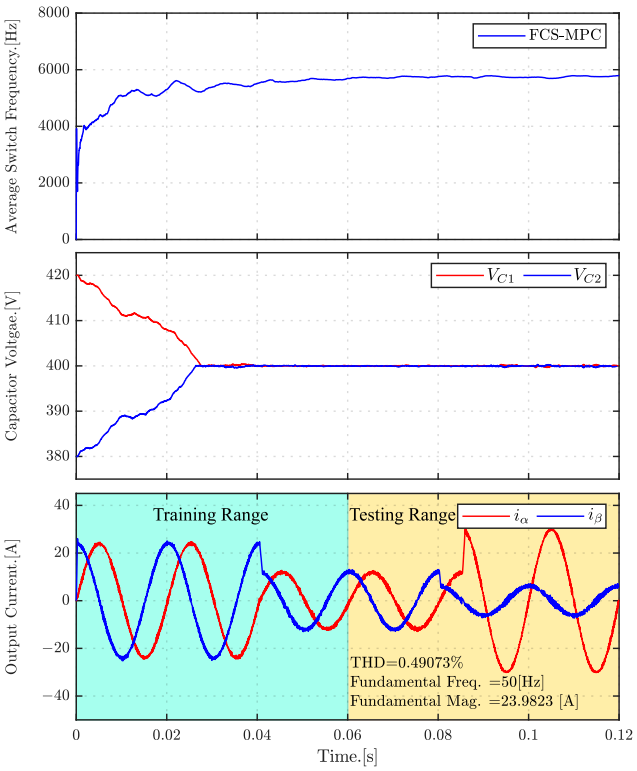

Fig. 9 FCS-MPC FIL simulation results for load current response applying decoupled-step on i_α^* and i_β^*

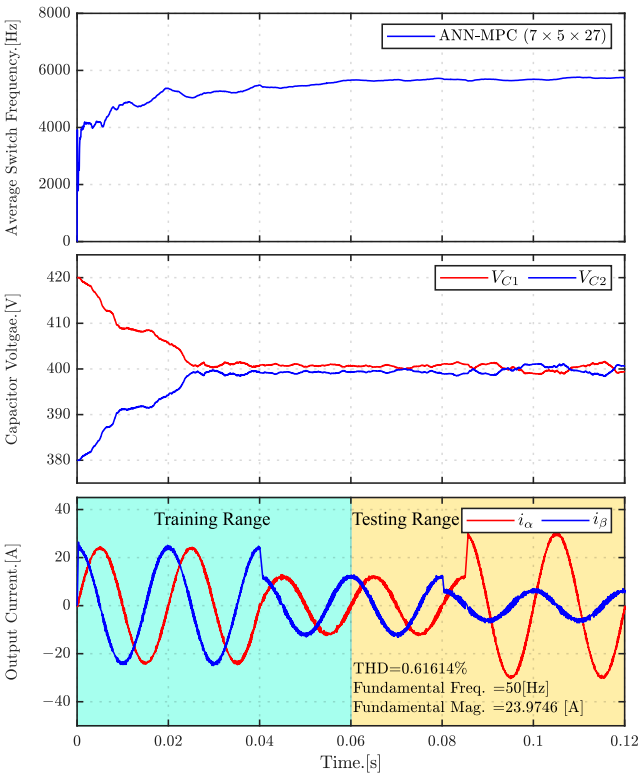

Fig. 10 ANN-MPC FIL simulation results for load current response applying decoupled-step on i_α^* and i_β^*

are equal to 3), respectively.

For long term operation, some of the system parameters would gradually change which motivates a large number of adaptive control method. Therefore, FLOPs operation containing only system parameters needs remaining. According to discrete model of the three-phase three-level NPC converter system in this article, the number of FLOPs of prediction model $n_{prediction}$ is 24, and that of cost function $n_{cost\ function}$ is 8, and the number of candidate switch state for one control period is 27, which means total number of FLOPs $N_{\mathrm{FCS-MPC}}^{NPC}$ is 864. On the other hand, the number of FLOPs of the designed ANN-MPC shown in Fig. 6 is $320 + 5\varepsilon_1 + 27\varepsilon_2$ (5 neurons in the hidden layer), which means much reduction of computation burden.

4.3 Selection of Number of Hidden Neurons

As mentioned in Section 3, till now there is no systematic method to obtain the best number of neurons in the hidden layer. Fig. 8 shows some statistics about the imitation accuracy for 4 ANN-MPCs with different number of neurons. It is obvious that the improved accuracy mostly come from better selection for redundant vectors as the increase of neuron amounts in the hidden layer. However, the cost performance (marginal benefit, net increasing accuracy per added neuron) decreases. In another word, more computation burden is required for same net increased accuracy as the increase of neuron amounts. On the other hand, the non-redundant voltage vector selection error comes from gradient descent training method itself. Because of the loss function definition, the most frequent appearing input states are more cared by the network, especially when the network is lightweight. As the results, there would be trade-off consideration between imitation accuracy and the computation burden.

4.4 Control Performance

Hardware description language (HDL) codes of both ANN-MPC and FCS-MPC are conducted by FIL implementation in MATLAB/Simulink environment, and the system parameters are all listed in the Table 1, and the structure details of ANN-MPC is listed in the Table 2. After enough offline training progress covering as wide as possible work range (the blue area in Fig. 9 and Fig. 10), the ANN-MPC directly replace FCS-MPC for online operation even facing unknown situation in training set (the red area).

The control performance lasting several periods for FCS-MPC and ANN-MPC are shown in Fig. 9 and Fig. 10, respectively. An initial voltage difference for the DC-link is set for voltage balance ability. And there is a reference current step drop at $t = 0.04\ s$ for transient state test. It is obvious that both MPC methods are highly similar with each other with only bit differences. The difference of THD between them is only around 0.1%, and the amplitude

deviation of ANN-MPC is 0.01 A bigger. Both of them achieve good voltage balance ability while ANN-MPC's dynamic response is even faster.

To show the decoupling between current control in $\alpha\beta$ frame, the amplitude of i_α^* (real component of reference current) was alone increased from 12 to 24 A at time $t = 0.08\,s$. It is clear that ANN-MPC shows comparable tracking ability for decoupled reference currents.

It is observed that, during ANN-MPC operation steady state, more ripples happen for the DC-link voltage balance. At the same time, the switching frequency is a little bit higher than that of FCS-MPC. According to the Fig. 8, the reason is that most of the imitation failures are for redundant voltage vectors selection, which means imitation resolution top limit of ANN.

5 Conclusion

In this article, the effectiveness of ANN-MPC strategy is verified by FPGA-in-the-loop (FIL) implementation of popularly used NPC converters in MATLAB/Simulink environment, which shows promising potentials to maintain advantages of advanced control technologies but reduce much computation complexity. The practical implementation of the ANN-based controller would be continued in pure hardware platform. Different from most image recognition applications, according to unique work requirement of power electronics systems, like high operation frequency, such Machine Learning (ML)-supported power electronics controllers would be tailored in future research.

Acknowledgments

This work was supported by the research project "Development of 20kW electric drive platform and integrated vehicle control module technology for commercialization" funded by the Ministry of Agriculture, Food and Rural Affairs of the Korean Government.

References

1. P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, J. Rodriguez: Predictive Control in Power Electronics and Drives. IEEE Transactions on Industrial Electronics, 55(12), 4312-4324 (2008)
2. J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, C. A. Rojas: State of the Art of Finite Control Set Model Predictive Control in Power Electronics. IEEE Transactions on Industrial Informatics, 9(2), 1003–1016 (2013)
3. R. Mikail, I. Husain, Y. Sozer, M. S. Islam, T. Sebastian: A Fixed Switching Frequency Predictive Current Control Method for Switched Reluctance Machines. IEEE Transactions on Industry Applications, 50(6), 3717–3726 (2014)
4. X. Yang, X. Liu, Z. Zhang, C. Garcia, J. Rodríguez: Two Effective Spectrum-Shaped FCS-MPC Approaches for Three-Level Neutral-Point-Clamped Power Converters. 2020 IEEE 9th International Power Electronics and Motion Control Conference (IPEMC2020-ECCE Asia), 1011-1016 (2020)
5. Zhao, S., Blaabjerg, F., Wang, H., Member, S: An Overview of Artificial Intelligence Applications for Power Electronics. IEEE Transactions on Power Electronics, 36(4), 4633–4658 (2021)
6. B. K. Bose: Artificial Intelligence Techniques: How Can it Solve Problems in Power Electronics?: An Advancing Frontier. IEEE Power Electronics Magazine, 7(4), 19-27 (2020)
7. K. Hornik, M. B. Stinchcombe, H. L. White: Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359-366 (1989)
8. T. Guillod, P. Papamanolis, J. W. Kolar: Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design. IEEE Open Journal of Power Electronics, 1, 284-299 (2020)
9. D. Chiozzi, M. Bernardoni, N. Delmonte, P. Cova: A Neural Network Based Approach to Simulate Electrothermal Device Interaction in SPICE Environment. IEEE Transactions on Power Electronics, 34(5), 4703-4710 (2019)
10. T. Dragičević, P. Wheeler, F. Blaabjerg: Artificial Intelligence Aided Automated Design for Reliability of Power Electronic Systems. IEEE Transactions on Power Electronics, 34(8), 7161-7171 (2019)
11. H. Soliman, P. Davari, H. Wang, F. Blaabjerg: Capacitance estimation algorithm based on DC-link voltage harmonics using artificial neural network in three-phase motor drive systems. 2017 IEEE Energy Conversion Congress and Exposition (ECCE), 5795-5802 (2017)
12. S. Lucia, D. Navarro, B. Karg, H. Sarnago, Ó. Lucía: Deep Learning-Based Model Predictive Control for Resonant Power Converters. IEEE Transactions on Industrial Informatics, 17(1), 409-420 (2021)
13. I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, A. A. Z. Diab: A Neural-Network-Based Model Predictive Control of Three-Phase Inverter With an Output LC Filter. IEEE Access, 7, 124737-124749 (2019)
14. D. S. Sarali, V. Agnes Idhaya Selvi, K. Pandiyan: An Improved Design for Neural-Network-Based Model Predictive Control of Three-Phase Inverters. 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable

Development (INCCES), 1-5 (2019)
15. D. Wang et al.: Model Predictive Control Using Artificial Neural Network for Power Converters. IEEE Transactions on Industrial Electronics, 69(4), 3689-3699 (2022)
16. S. Wang, T. Dragicevic, Y. Gao, R. Teodorescu: Neural Network Based Model Predictive Controllers for Modular Multilevel Converters. IEEE Transactions on Energy Conversion, 36(2), 1562-1571 (2021)
17. M. Novak, T. Dragicevic: Supervised Imitation Learning of Finite-Set Model Predictive Control Systems for Power Electronics. IEEE Transactions on Industrial Electronics, 68(2), 1717-1723 (2021)
18. F. Grimm, P. Kolahian, Z. Zhang, M. Baghdadi: A Sphere Decoding Algorithm for Multistep Sequential Model-Predictive Control. IEEE Transactions on Industry Applications, 57(3), 2931-2940 (2021)
19. J. He, L. Xing, C. Wen: Weighting Factors' Real-time Updating for Finite Control Set Model Predictive Control of Power Converters via Reinforcement Learning. 2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA), 707-712 (2021)
20. P. Zanchetta: Heuristic multi-objective optimization for cost function weights selection in finite states model predictive control. 2011 Workshop on Predictive Control of Electrical Drives and Power Electronics, 70-75 (2011)