Weighting factor design based on Artificial Neural Network for Finite Set MPC operated 3L-NPC converter

Novak, Mateja; Dragicevic, Tomislav; Blaabjerg, Frede

Published in:
34th Annual IEEE Applied Power Electronics Conference and Exposition, APEC 2019

DOI (link to publication from Publisher):
10.1109/APEC.2019.8722062

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Weighting factor design based on Artificial Neural Network for Finite Set MPC operated 3L-NPC converter

Mateja Novak
Department of Energy Technology
Aalborg University
Aalborg, Denmark
Email: nov@et.aau.dk

Tomislav Dragicevic
Department of Energy Technology
Aalborg University
Aalborg, Denmark
Email: tdr@et.aau.dk

Frede Blaabjerg
Department of Energy Technology
Aalborg University
Aalborg, Denmark
Email: fbl@et.aau.dk

***Abstract*—Optimum design of the weighting factors for a multi-objective cost function is one of the major challenges of Finite-Set Model Predictive Control (FS-MPC) operated power electronic converters. Especially for multi-level topologies, where multi-objectives must be included in the cost function to ensure a safe operation of the converter, the complexity of the optimization problem is rapidly growing with each new objective included in the cost function. In this paper a new approach for design of the weighting factors for a three level neutral point clamped (NPC) converter using artificial neural network (ANN) is proposed. The ANN is used as a surrogate model of the detailed converter model. In the first step a detailed converter model is simulated for different weighting factor combinations. From the simulations obtained performance metrics (e.g. total harmonic distortion (THD), average switching frequency, DC-link voltage balance) are used to train the ANN. Once the network is trained, it can be used to estimate the performance metrics for any combination of weighting factors. By defining a fitness function using the metrics, weighting factor combinations that optimize the function are found to be very fast. The design is also validated on an experimental set-up, where the measured performance metrics are compared to the ones predicted by the ANN. It is concluded that the results match very well with a difference being below 10%.**

I. Introduction

Due to the increase of computational power of microprocessors a large number of advanced control algorithms for power electronic converters has been proposed over the past years. New algorithms are aiming to overcome the limitations of the well known classical control algorithms based on linear cascaded loops. Especially for multilevel converter topologies like neutral point clamped (NPC) converters, algorithms that can easily include multiple objectives are becoming attractive alternatives. Finite set model predictive control (FS-MPC) is one of them and it gained the popularity because of a straightforward inclusion of the objectives and a simple implementation [1]. Using the discrete model of the system, future values of the system voltages and currents are calculated for all possible converter switching combinations. The desired behavior of the converter is defined through a cost function and the balance of the objectives is set by weighting factors.

How to find the optimum weighting factor values is still an open research question [2]. The problem is not trivial as a trial and error approach can be very time consuming. There have been several attempts to solve this issue and some of them even proposed to remove the weighting factors [3]–[5]. In [3] it must be noticed that with the removal of the weighting factors, there still remain coefficients in the cost function that are chosen by a heuristic approach and the approach in [4] is only suitable for systems where there are no conflictive objectives. One of the first proposals to reduce the time consumption was to use the branch and bound search [6], but the approach was still too empirical. In [7], [8] methods for online adaptation of the weighting factors were proposed. However, an online approach could impose a large computational burden for evaluation of complex cost-functions. Another attempt was made using a genetic algorithm optimization to find the two optimum weighting factors [9]. The drawback of this offline method is that each design objective needs a new set of simulations.

A good defined approach for weighting factor design should offer the user a way to easily obtain an optimum system performance for a design criteria like low THD, low switching frequency etc. and not present a heavy computational burden for the algorithm application. In this paper we are proposing to use artificial neural networks (ANN) to automate this procedure and to obtain the optimum weighting factors. The method is used offline and does not require any modifications of the cost function nor is it increasing the computational burden. Once the ANN is trained it can be used to find the optimum weighting factors for any design criteria and the result will be calculated in just a few seconds. The training data collection from a detailed simulation model can easily be parallelized and automatized. Therefore the execution on a multi-core computer can be very fast. The collection of the training data needs to be performed once and not several times like in [9] if a different design objective is selected. The criteria for the weighting factor design is defined in a fitness function using the performance metrics, which are chosen depending on the converter application and topology e.g. for the two-level voltage source converter topology the THD of the output voltage or current and switching frequency could be sufficient but for the multilevel topologies like NPC, the balance of the DC-link voltages should also be included.

The paper is structured as follows: in the first part the system model along with the control algorithm is introduced. Section III explains the proposed ANN based weighting factor design method and Section IV presents the application on a 3L-NPC converter. For the obtained weighting factor combinations, ANN performance metrics, detailed simulation model metrics and metrics from experimental results are compared. At the end of the paper conclusions and future work aspects are given.

II. System Model

The FS-MPC algorithm is using the system model of the configuration shown in Fig. 1 to predict the behavior of the system for all 27 possible switching combinations in a 3L-NPC converter. The presented configuration is typically used for Uninterruptible Power Supplies (UPS) where the main objective lies on high quality of the output voltage. In every sample step new measurements of the filter and DC-link capacitor voltages and filter currents are obtained to calculate the system predictions using the differential equations which capture the DC and the AC side dynamics of the system.

On the DC-side predictions of the DC-link capacitor voltages and charging currents are calculated using the following equations:

$$v_{dc1,2}(t) = C_{dc1,2}\frac{di_{dc1,2}(t)}{dt} \quad (1)$$

$$i_{dc1,2}(t) = i_{dc}(t) \mp \sum_{x=a,b,c} H_{1,2x} \cdot i_{fx}(t) \quad (2)$$

where $v_{dc1,2}(t)$ are voltages across the DC-link capacitors $C_{dc1,2}$ and $i_{dc1,2}(t)$ are the respective charging currents. $i_{f\,abc}(t)$ are the inverter phase currents and $i_{dc}(t)$ is the DC source current. H_{1x} and H_{2x} are indicator functions with the following logic: H_{1x} will return 1 if the phase leg $x \in a, b, c$ is connected to $V_{dc}/2$ while H_{2x} returns 1 if the phase leg is connected to $-V_{dc}/2$, otherwise the function values are 0.

On the AC side LC output filter equations in the stationary $\alpha\beta$ frame are used to predict the filter voltages and currents:

$$v_{i\,\alpha\beta}(t) = L_f\frac{di_{f\,\alpha\beta}(t)}{dt} + v_{c\,\alpha\beta}(t) \quad (3)$$

$$i_{f\,\alpha\beta}(t) = C_f\frac{dv_{c\,\alpha\beta}(t)}{dt} + i_{o\,\alpha\beta}(t) \quad (4)$$

where $i_{o\,\alpha\beta}$ are the load currents, $v_{c\,\alpha\beta}$ and $v_{i\,\alpha\beta}$ are filter and inverter output voltages, L_f and C_f are filter inductance and capacitance. Using the Euler forward method as presented in [10] the equations are discretized and used to calculate the future states of system voltages and currents. For a safe operation of the 3L-NPC converter in this configuration two objectives must be included in the algorithm cost function: minimization of the reference tracking error and neutral point voltage balancing (g_{dc}). In this example which will further be used for application of the ANN weighting factor design

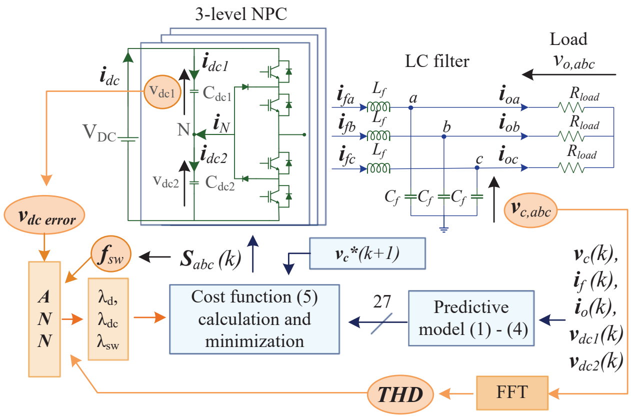


Fig. 1. Schematics of a 3L-NPC converter system for UPS.

approach, two additional terms will be included in the cost function: a derivative term to further improve the reference tracking by taking into account the heading of the capacitor voltage trajectory as demonstrated in [10] and minimization of the switching frequency (g_{sw}). The latter part is achieved by comparing the previous $S_x(k-1)$ and current $S_x(k)$ switching state for all converter phase legs $x \in a, b, c$. These objectives will be included in the algorithm as follows:

$$g = (v_{c\alpha}^* - v_{c\alpha}^P)^2 + (v_{c\beta}^* - v_{c\beta}^P)^2 + \lambda_d \cdot g_d + \lambda_{dc} \cdot g_{dc} + \lambda_{sw} \cdot g_{sw} \quad (5)$$

$$g_d = (i_{f\alpha}^P - i_{o\alpha} + C_f \cdot \omega_{ref} \cdot v_\beta^*)^2 + (i_{f\beta}^P - i_{o\beta} - C_f \cdot \omega_{ref} \cdot v_\alpha^*)^2 \quad (6)$$

$$g_{dc} = (v_{dc1}^P - v_{dc2}^P)^2 \quad (7)$$

$$g_{sw} = \sum_{x=a,b,c} |S_x(k) - S_x(k-1)|, \quad (8)$$

where $v_{c\,\alpha\beta}^P$, $i_{f\alpha\beta}^P$ and $v_{dc1,2}^P$ are the predicted values of the filter voltage, filter current and DC-link voltages; and $v_{c\,\alpha\beta}^*$ are the reference values of the filter voltage. As it can be noted three weighting factors are present in this cost function: λ_d, λ_{dc} and λ_{sw} and each of them defines the importance of different objective. To illustrate the complexity of the optimization problem, if for example the weighting factor range for the three factors is chosen from 0 to 5 with a step of 0.5 to cover the practical design range, 11 different values are possible for one factor which in total results in 1331 different combinations. It is clear that if for example a branch and bound method [6] would be used to solve this optimization problem the process would be too time consuming and that a non-empirical method should be used to solve this problem. Therefore, in the following section a solution for this problem based on ANN will be presented. We don't expect a higher problem complexity because the highest number of objectives in the cost functions, presented in the recent publications for power electronics applications, is rarely larger than two [2].

III. WEIGHTING FACTOR DESIGN

In this section the concept of weighting factor design using the ANN will be explained. Three stages in the process of obtaining the optimal weighting factors can be noticed:

1) Performance metrics data set collection from a detailed simulation model.
2) ANN training using the obtained data sets.
3) Fitness function minimization using the trained ANN model.

In the first stage for numerous weighting factor combinations λ_d, λ_{dc} and λ_{sw} the performance metrics: THD of the filter voltage, average switching frequency $f_{sw\,avg}$, DC-link capacitor voltage ripple $v_{dc\,rip}$ and error $v_{dc\,error}$ are obtained from the simulations of the detailed system model. This step can easily be executed on a multi-core computer using the parallel computing tool in MATLAB to speed up the process. It is important to notice a trade-off between the future ANN model precision i.e. the number of data samples and the execution time. More samples will lead to a more precise model, however more computing resources will be needed to obtain the data in a reasonable time.

The data-sets are then normalized and used in the next step to train the ANN, which afterwards can be used to quickly obtain the performance metrics for any combination of the weighting factors. The speed of this process is several magnitudes higher compared to the simulations of the detailed model. Also these two steps need to be done just once for the set-up model parameters. Because of the static relationship between the input and output data, a feed-forward type ANN was selected. In Fig. 2 an example of a feed-forward ANN is shown. Three types of layers can be identified from the figure: input layer which has 3 neurons, a hidden layer and an output layer with 3 neurons. In every layer the neurons process the information received from the lower layer. The outputs of a neuron is calculated using the outputs of all layers below multiplied by associated weights and summed together with a bias term. The process of adjusting the parameters of the ANN (weights and bias terms) is done using back-propagation algorithm [11]. This algorithm is implemented in standard softwares like MATLAB's Neural Networks toolbox, which will be used in the next section. For more information about the feed forward ANN the reader is referred to the following references [11], [12]. In the last step of the design, the trained ANN is used to evaluate the user-defined fitness function f_{ANN}. The f_{ANN} is a combination of performance metrics, where the user defines the preference of the low THD, low switching frequency or low DC-link voltage ripple.

IV. APPLICATION ON A 3L-NPC CONVERTER

In this section the application of the ANN based weighting factor design will be presented on a 3L-NPC converter for UPS application. For the application on a 2L-VSC topology the reader is referred to [13] where more information is given about which metrics to use for this application and how the ANN can be structured. Using MATLAB/Simulink a detailed simulation model with the parameters presented in the Table I was created. The parameters are equivalent to the experimental set-up, which will be used for design validation. Converter dead time T_d = 3 μs and one step computational delay were also included in the simulation model in order to reproduce the conditions in the experimental set-up.

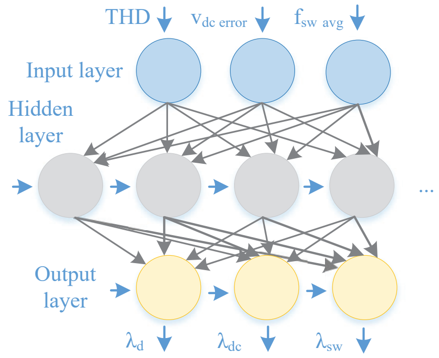


Fig. 2. Schematics of a feed-forward artificial neural networks (ANN) structure in Fig. 1.

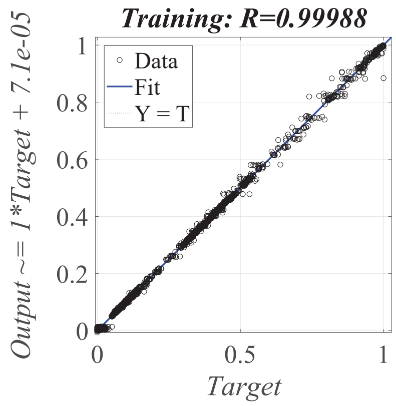


Fig. 3. Regression plot for the performed ANN training showing the relationship between the training data (o) and predicted data (blue line).

TABLE I: System parameters used for testing.

Parameter	Value
DC-link voltage (V_{DC})	520 V
DC-link capacitors (C_{dc1}, C_{dc2})	4 mF
Output filter inductance (L_f)	2.4 mH
Output filter capacitance (C_f)	15 μF
Load resistance (R_{load})	60 Ω
Reference voltage and frequency ($V_c^*, f_{v_c}^*$)	230 V, 50 Hz
Sampling time (T_s)	25 μ s

A. ANN training

For a range from 0 to 5 with a step of 0.5, weighting factors λ_d, λ_{dc} and λ_{sw} were varied in every simulation. Altogether $11^3 = 1331$ different combinations were simulated. The duration of each simulation was set to 0.5 s. During the testing it was noticed that if a shorter simulation time was used some initial transients in the DC-link capacitor voltages were captured, which resulted in very random data-sets of the measured DC-link ripple and error. Consequently the precision of the trained ANN was lower. Therefore, the measurements from the time interval 0.3 to 0.5 s (10 cycles) were used for further evaluation. In this period Fast Fourier Transform (FFT) algorithm from the SimPowerSystems toolbox was used to calculate the THD of the filter voltage and the DC-link voltage error was selected to capture the performance of the neutral point balancing.

The average switching frequency was calculated using the following expression:

$$f_{sw_{avg}} = \sum_{i=1}^{8000} \frac{|\Delta S_a(i)|+|\Delta S_a(i)|+|\Delta S_a(i)|}{12}, \tag{9}$$

where $\Delta S_x(i)$ represents the number of switches that changed the switching state in each phase leg. The summation is performed until 8000 as this is the number of samples in the time interval with the sample time $T_s = 25\mu s$. The simulations were performed using the parallel computing toolbox from MATLAB. Because the duration of simulation was long, a multi-core computer with 24 cores was used to obtain the results.

In the next step the obtained data sets were used to train the ANN with 3 neurons in the input layer, 2 hidden layers with 10 and 5 neurons and the output layer with 3 neurons. The training was performed using the Neural Networks toolbox in MATLAB and the training performance can be seen in Fig. 3. The toolbox provides several plots that can be used to evaluate the performance of the training. One of those plots is the regression plot, which shows the relationship between the output and input data. It can be observed that the data points fit the linear regression line. Another indicator of the relationship between the outputs and targets that the toolbox provides is the R value, which is close to 1 meaning the relationship is almost linear. The ANN training is therefore successfully completed.

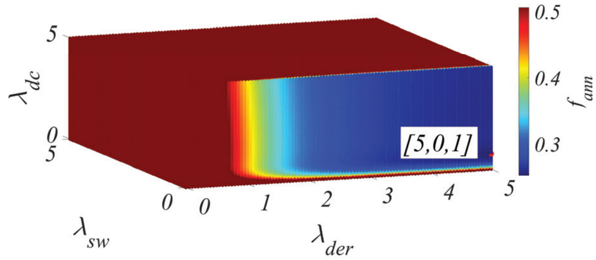
B. Fitness functions

In the next step a fitness function using the performance metrics can be defined. To demonstrate the approach, following two fitness functions will be used:

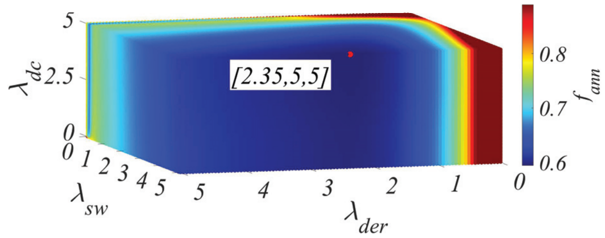
$$f_{ANN} = THD^2_{ANN} + 1/v^2_{dc\,ripANN} \tag{10}$$

$$f_{ANN} = THD^2_{ANN} + 0.5 f_{sw\,ANN} + 1/v^2_{dc\,ripANN} \tag{11}$$

The objective in the first fitness function is to find the optimum weighting factor combination to produce the minimal output voltage THD and a good balancing of the DC-link voltages $v_{dc\,1,2}$ regardless of the converter switching frequency. In the second fitness function a third objective is added, minimization of the switching frequency f_{ANN}.



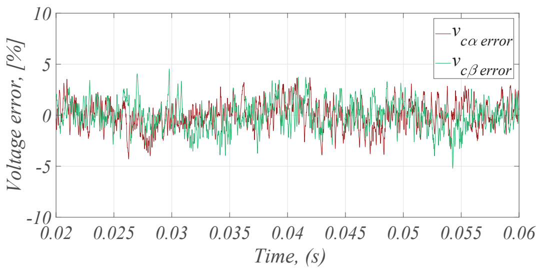
(a) Plot of the fitness function (10) and found optimal weighting factors: $\lambda_d = 5, \lambda_{dc} = 1$ and $\lambda_{sw} = 0$.



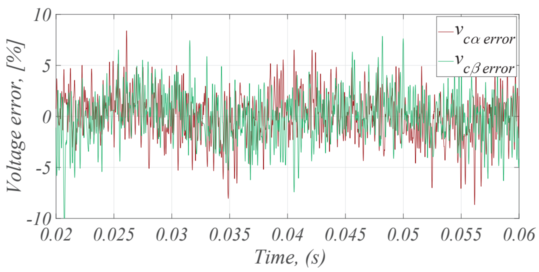
(b) Plot of the fitness function (11) and found optimal weighting factors: $\lambda_d = 2.35, \lambda_{dc} = 5$ and $\lambda_{sw} = 5$.

Fig. 4. Results of the ANN training for the two fitness functions and system parameters in Table I.

The results are shown in the Fig. 4a and Fig. 4b with the following optimum weighting factors obtained as a minimum of (10): $\lambda_d = 5, \lambda_{dc} = 1$ and $\lambda_{sw} = 0$; $\lambda_d = 2.35$, $\lambda_{dc} = 5$ and $\lambda_{sw} = 5$ for the (11). The minimization of the fitness function (10) resulted with a $\lambda_{sw} = 0$, which is exactly what was expected as we didn't include the $f_{sw\,ANN}$ in the fitness function and gave the higher priority to achieve a low THD. This can also be noticed in the selection of the λ_d which was selected as maximum of the weighting factor range and also in the plot of the fitness function in Fig. 4a where all combinations with the switching frequency minimization objective produced a large f_{ANN} value. It is important also to notice that the $v_{dc\,ripANN}$ always needs to be included in the fitness function as otherwise combinations that have a low THD and highly unbalanced $v_{dc\,1,2}$ may be selected and in the extreme cases even the options without any balance ($\lambda_{dc} = 0$). In Fig. 4b for (11) a very high value of the fitness function was calculated for combinations with a high λ_{sw} and low λ_{dc} while the combinations of higher values resulted in minimal f_{ANN} values. A high λ_{dc} was computed as a result of DC-link balancing and minimization of switching frequency being opposing control objectives. Therefore, if we want to obtain a low switching frequency we also need to increase the weighing factor of the DC-link balancing objective to get a good balance of the DC-link voltages. In case the λ_{sw} is much larger than the λ_{dc} it is not possible to safely operate the system because the DC-link balance can not be established. Compared to the



(a) Voltage reference tracking error waveform in $\alpha\beta$ reference plane with weighting factors $\lambda_d = 5, \lambda_{dc} = 1$ and $\lambda_{sw} = 0$ that minimize the fitness function (10).



(b) Voltage reference tracking error waveform in $\alpha\beta$ reference plane with weighting factors $\lambda_d = 2.35, \lambda_{dc} = 5$ and $\lambda_{sw} = 5$ that minimize the fitness function (11).

Fig. 5. Reference tracking performance of the detailed Simulink model for two weighting factor sets.

first case, $\lambda_d = 2.35$ was here selected in the middle of the range as now the fitness function needs to find a combination of weighting factors that will produce a good balance of low THD and low switching frequency.

The voltage reference tracking error waveform in $\alpha\beta$ reference plane for the obtained weighting factor combinations for the fitness functions are presented in the Fig. 5a and Fig. 5b while the metrics predictions of the ANN for these weighting factors and metrics from a detailed simulation model can be found in Table II. It can be seen that the error for the ANN predicted THD is below 3.5% and the f_{sw} is below 1.5%.

C. Experimental validation

The weighting factor combinations were also validated on a Semikron 3L SKiiP28MLI07E3V1 Evaluation Inverter [14] with the control algorithm implemented on a MicroLabBox DS1202 PowerPC DualCore 2 GHz processor board from dSpace shown in the Fig. 6. In Fig. 7 and Fig. 8 the output filter voltage and the voltages on the two DC-link capacitors are shown. It can be noticed that both combinations produced a very good balancing of the DC voltages. From Table II a very good match with the experimental results can be observed. More weighting factor combinations were also compared and the difference to ANN was always below 10%.

V. Conclusion

A new method based on ANN for design the weighting factors for a FS-MPC controlled 3L-NPC converter was presented. Once trained ANN can very fast produce optimum weighting factor combinations for user-defined fitness functions. The approach is applicable to various converter applications. It does not impose an additional computational burden as the selection of the weighting factors is performed offline nor does it change the structure of the cost function. The process of collecting the data through the detailed simulation model can be automated and paralleled in order to save time. The predicted response from the ANN matched very well the detailed model results (less than 5% error) and a good match with experimental results was also observed (less than 10% error). In the future work the performance of the ANN

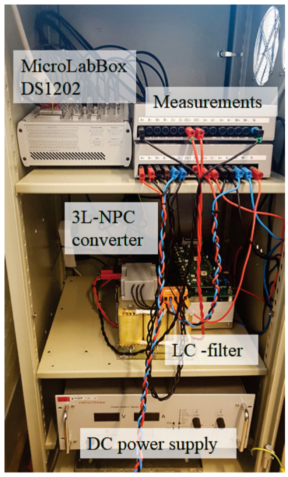
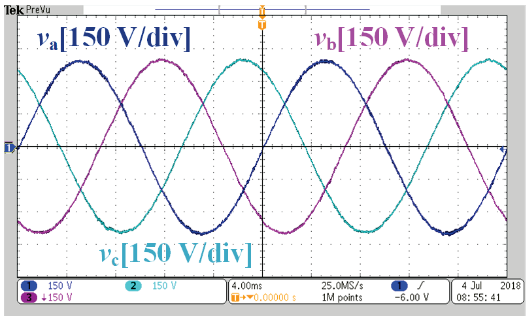
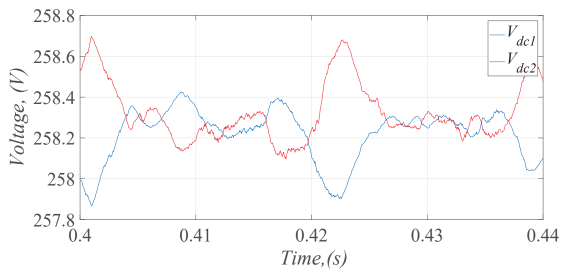


Fig. 6. 3L-NPC experimental set-up.



(a) Filter capacitor voltage $v_{c\,abc}$.

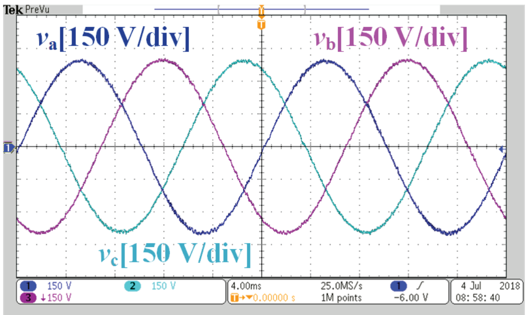


(b) DC-link capacitor voltages $v_{dc1,2}$.

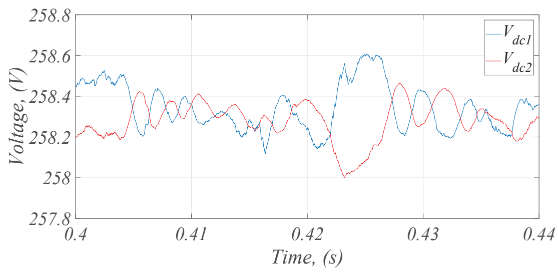
Fig. 7. Experimental results for the fitness function (10).

TABLE II: Comparison of metrics from ANN, detailed model and experimental set-up.

f_{ANN}	Metrics	ANN results	Detailed model results	Experimental results
$THD_{ANN}^2 + 1/v_{dc\,ripANN}^2$	THD	0.79 %	0.79 %	0.86%
$THD_{ANN}^2 + 1/v_{dc\,ripANN}^2$	f_{sw}	6 kHz	5.8 kHz	5.5 kHz
$THD_{ANN}^2 + 1/v_{dc\,ripANN}^2$	$v_{dc\,error}$	$<$ 1 V/period	$<$ 1 V/period	$<$ 1 V/period
$THD_{ANN}^2 + 0.5 f_{sw\,ANN} + 1/v_{dc\,ripANN}^2$	THD	1.26.%	1.3%	1.23%
$THD_{ANN}^2 + 0.5 f_{sw\,ANN} + 1/v_{dc\,ripANN}^2$	f_{sw}	2.2 kHz	2.19 kHz	2.3 kHz
$THD_{ANN}^2 + 0.5 f_{sw\,ANN} + 1/v_{dc\,ripANN}^2$	$v_{dc\,error}$	$<$ 1 V/period	$<$ 1 V/period	$<$ 1 V/period



(a) Filter capacitor voltage $v_{c\,abc}$.



(b) DC-link capacitor voltages $v_{dc1,2}$.

Fig. 8. Experimental results for the fitness function (11).

could be improved by training the ANN using the metrics data obtained from experiments. Then even a closer match with the experimental results could be achieved. It would also be interesting to explore in the future work if other types of ANN could produce an even better match to the detailed simulation model.

References

[1] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 16–31, March 2014.

[2] 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, Feb 2017.

[3] F. Villarroel, J. R. Espinoza, C. A. Rojas, J. Rodriguez, M. Rivera, and D. Sbarbaro, "Multiobjective switching state selector for finite-states model predictive control based on fuzzy decision making in a matrix converter," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 589–599, Feb 2013.

[4] E. Fuentes, C. A. Silva, and R. M. Kennel, "MPC implementation of a quasi-time-optimal speed control for a PMSM drive, with inner modulated-FS-MPC torque control," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3897–3905, June 2016.

[5] Y. Yang, H. Wen, M. Fan, M. Xie, and R. Chen, "A fast finite-switching-state model predictive control method without weighting factors for T-type three-level three-phase inverters," *IEEE Trans. Ind. Informatics*, pp. 1–12, 2018.

[6] P. Cortes, S. Kouro, B. L. Rocca, R. Vargas, J. Rodriguez, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Guidelines for weighting factors design in model predictive control of power converters and drives," in *2009 IEEE International Conference on Industrial Technology*, Feb 2009, pp. 1–7.

[7] S. A. Davari, D. A. Khaburi, and R. Kennel, "An improved FCS-MPC algorithm for an induction motor with an imposed optimized weighting factor," *IEEE Trans. Power Electron.*, vol. 27, no. 3, pp. 1540–1551, March 2012.

[8] Z. Zhang, W. Tian, W. Xiong, and R. Kennel, "Predictive torque control of induction machines fed by 3L-NPC converters with online weighting factor adjustment using fuzzy logic," in *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, June 2017, pp. 84–89.

[9] P. Zanchetta, "Heuristic multi-objective optimization for cost function weights selection in finite states model predictive control," in *2011 Workshop on Predictive Control of Electrical Drives and Power Electronics*, Oct 2011, pp. 70–75.

[10] T. Dragicevic, "Model predictive control of power converters for robust and fast operation of AC microgrids," *IEEE Trans. Power Electron.*, vol. 33, no. 7, pp. 6304–6317, July 2018.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359 – 366, 1989.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, 2014.

[13] T. Dragicevic and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Trans. Ind. Electron.*, pp. 1–12, In press.

[14] *3L SKiiP28MLI07E3V1 Evaluation Inverter Technical Explanation*, Semikron, 2014, rev. 2.0.