Supervised imitation learning of finite set model predictive control systems for power electronics

Novak, Mateja; Dragicevic, Tomislav

Published in:
I E E E Transactions on Industrial Electronics

DOI (link to publication from Publisher):
10.1109/TIE.2020.2969116

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Supervised imitation learning of finite set model predictive control systems for power electronics

Mateja Novak, *Student Member, IEEE*, and Tomislav Dragicevic, *Senior Member, IEEE*

***Abstract*—In the past years finite set model predictive control (FS-MPC) has received a lot of attention in the power electronics field. Due to very simple inclusion of the control objectives and straightforward design, it has been adopted in a lot of different converter topologies. However, computational burden often imposes limitations in the control implementation if multistep predictions are deployed or/and if multilevel converters with many possible switching states are used. To remove these limitations, we propose to imitate the predictive controller. It is important to highlight that the imitator is not intended to improve the dynamic or steady-state performance of the original FS-MPC algorithm. In contrast, its key role is to keep approximately the same performance while at the same time reducing the computational burden. Our proposed imitator is an artificial neural network (ANN) trained offline using data labelled by the original FS-MPC algorithm. Since the computational burden of the imitator is not correlated with the complexity of the FS-MPC algorithm it emulates, implementation of much more complex predictive controllers is made possible without prior limitations. The proposed method has been validated experimentally on a stand-alone converter configuration and the results have confirmed a good match between the imitator and the predictive controller performance. Simulation models of both controllers are provided in the supplementary files for three different prediction horizons.**

***Index Terms*—Artificial neural networks, control design, DC-AC converters, finite-set model predictive control, supervised imitation learning.**

I. INTRODUCTION

THE idea of using the artificial neural networks (ANN) for power electronic control systems was considered already in the early 90's [1]–[3]. One of the main reasons why this idea was pursued is the capability of ANN to process complex signals in a short time. Due to its parallel structure, the computational burden was lower then for the conventional controllers which were limiting the switching frequency of the converters. Moreover, the idea of having a controller that could adapt to the changing environment conditions was considered very attractive. Later, with further advances in microprocessor technology, more ANN based controllers were proposed in power electronic control systems [4]–[6].

At the same time in the control design, a combination of ANN with model predictive control (MPC) algorithm gained a lot of interest. ANN can be used for solving the optimization problem of the MPC [7] or to create a system model [8], [9]. Furthermore, MPC can be used as a supervisor in the imitation learning settings like it was presented in [10] for control of the quadrotors or it can collect the data used for training the NN policy of an autonomous aerial vehicle in [11]. Applications where NN are used for synthesis of the MPC are presented in [12] for robust MPC, for conventional MPC in [13] and for finite-set MPC in [14]. In [12] the approximator implementation through NN was evaluated to be 200 timed faster than the original algorithm, showing how this approach can enable computationally efficient implementations of MPC algorithms. However, the presented approach was only evaluated on a numerical example [12] and it was not experimentally validated [13], [14].

In this paper we will use the ANN as a controller imitator for the finite-set model predictive control (FS-MPC). FS-MPC is an advanced control method that has gained a lot of attention in recent years due to its straightforward design, simple inclusion of different objectives and its discrete nature that is natural for control of power converters [15]. Thus, it was not a surprise that FS-MPC has been adapted very quickly to various topologies of power electronic converters [16]. However, there is still one limitation that is stopping the potential of FS-MPC in power electronics. Namely, multi-step horizon prediction algorithms or the applications on multilevel or multicell converters are still limited due to the large computational burden. As the FS-MPC has to perform a number of iterative calculations to find the optimum voltage vector, the more voltage vectors one topology has, the more calculations are needed. Similarly, for multi-step prediction horizons, the number of candidate switching options is increasing exponentially.

Therefore, many simplifications in the form of sorting algorithms, extrapolations and cost function modifications were necessary in practical implementations to reduce the number of candidate switching states. In [17] the authors have proposed a reduced complexity FS-MPC for multilevel converter topologies where DC-link voltage balancing was removed from the cost function and in [18] the authors have split the control objectives in two stages for modular multi-level converter (MMC) application. These solutions are sacrificing performance to reduce the computation burden. Simplifications necessary for reducing the candidate switching states for prediction and evaluation in the cost function for a matrix converter are presented in [19]. A sorting algorithm is another way to reduce the number of the candidate switch-

Manuscript received August 23, 2019; revised November 5 and December 10, 2019; accepted December 23, 2019.

M. Novak and T. Dragicevic are with the Department of Energy Technology, Aalborg University, Aalborg, Denmark.

ing options in MMC applications [20]. Multi-step prediction horizon algorithms have shown a great potential in drives application but they can not be implemented without sorting algorithms and extrapolations to reduce the complexity [21], [22].

The approach presented in this paper does not require modifications of the conventional FS-MPC algorithm or design of a heuristic search algorithm. Using the ANN, an accurate imitator of the conventional FS-MPC algorithm can be created. It needs to be mentioned that the imitator is not intended to outperform the original controller. As it will be shown experimentally, the key advantage of the imitator compared to standard controllers is that its execution time is independent of the complexity and prediction horizon of imitated controller. Namely, it depends only on the number of neurons in the network. Compared to the deep neural network approach in [6], in our approach only one hidden layer is necessary to control all the switches in the converter topology and the imitator accuracy is much higher. This is confirmed both in simulations and experiments. We also present a novel data generation process, which allows a much quicker and comprehensive covering of the controller's actuation space then in [14]. It needs to be mentioned that the presented simulation results in [14] show that the performance of the trained imitator controller is much better than the performance of the FS-MPC algorithm. This is only possible if the imitator controller is switching with a higher switching frequency. Indeed, the authors have not shown the calculated switching frequency to confirm that the two controllers operate with approximately the same switching frequency. Therefore, the controller in [14] is actually not applying the same control strategy as the original controller and it is hence not fair to classify it as an imitator. Additionally, the method was only introduced for one step-prediction horizon, without delay compensation included in the data generation, which makes it impossible to implement in the experimental platforms and validate experimentally. In our opinion, contributions of this letter have many practical aspects, which the authors in [12]–[14] did not address.

II. System model

Supervised imitation learning approach in this paper is demonstrated on a stand-alone VSC with an LC output filter and a resistive load. This converter configuration can typically be found in the uninterruptible power supply (UPS) systems and AC microgrids. However, it is worth noting that proposed approach is general and can be applied to imitate any type of controller and any converter topology. The control algorithm needs to maintain a smooth sinusoidal output voltage and give a fast response under load variations. In Fig. 1, it is shown that for the implementation of the imitator controller, it is necessary to measure the filter currents $i_{f\,abc}$, capacitor voltages $v_{c\,abc}$ and load currents $i_{o\,abc}$. These measurements are also necessary for implementation of the conventional FS-MPC algorithm. Together with the previously applied voltage vector, these measurements are used to generate the input data vector for the pattern recognition neural network.

The imitator controller will learn to imitate the conventional FS-MPC algorithm presented in [23]. The FS-MPC algorithm uses the differential equations of the filter currents and capacitor voltages to calculate the future propagations of these variables:

$$\begin{bmatrix} \frac{di_f(t)}{dt} \\ \frac{dv_c(t)}{dt} \end{bmatrix} = \mathbf{A} \begin{bmatrix} i_f(t) \\ v_c(t) \end{bmatrix} + \mathbf{B} \begin{bmatrix} v_i(t) \\ i_o(t) \end{bmatrix} \tag{1}$$

where

$$\mathbf{A} = \begin{bmatrix} -\frac{R_f}{L_f} & -\frac{1}{L_f} \\ \frac{1}{C_f} & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{L_f} & 0 \\ 0 & -\frac{1}{C_f} \end{bmatrix} \tag{2}$$

Before the implementation, the equations are discretized using the Euler forward discretization method with the sampling period of T_s = 20 μs. Following cost function is used in the algorithm:

$$\begin{aligned} g &= \left(v_{f\alpha}^* - v_{f\alpha}\right)^2 + \left(v_{f\beta}^* - v_{f\beta}\right)^2 + \lambda_d \cdot g_d + h_{lim} \\ g_d &= \left(C_f \omega v_{f\beta}^* - i_{f\alpha} + i_{o\alpha}\right)^2 + \left(C_f \omega v_{f\alpha}^* + i_{f\beta} - i_{o\beta}\right)^2 \\ h_{lim} &= \begin{cases} 0, \text{ if } |\bar{i}_f| \leq i_{max} \\ \infty, \text{ if } |\bar{i}_f| > i_{max} \end{cases} \end{aligned} \tag{3}$$

where $\bar{v}_f^* = v_{f\alpha}^* + jv_{f\beta}^*$ is the voltage reference vector, ω is the reference frequency and λ_d is the weighting factor of the additional current reference term. This term was proposed in [24] for improving the steady state performance. h_{lim} is limiting the inductor current to the i_{max} = 30 A, which is the nominal current of the devices.

Several approaches for weighting factor design can be found in the literature. The simplest is the branch and bound algorithm proposed by authors in [25] and this approach is sufficient for selecting the one weight used in the cost function (3). More advanced approaches based on genetic algorithms, fuzzy logic or ANN are available for higher complexity optimization problems, where two or more weighting factors are included in the cost function [11], [26], [27]. In the example used in the manuscript the criteria for selecting the weighting factor λ_d was the THD of the filter capacitor voltage. The threshold was set to 1.5%, which is in accordance with the IEC 62040-3 standard for UPS applications [28]. For λ_d = 0 the obtained THD is 1.79%, when the weighting factor is increased to λ_d = 1, the THD value is decreased to 1.33%. The weighting factor could also be further increased, however that would also cause the switching frequency and the switching losses to increase. The limit for the switching frequency was set to 8 kHz. Table I shows system parameters used in the control algorithm.

TABLE I: System parameters

Parameter	Value
DC-link voltage	V_{dc} = 700 V
Output filter	L_f = 2.4 mH, R_f = 0.1 Ω, C_f = 14.2 μF
Algorithm sampling time	T_s = 20 μs
Reference frequency	f_{ref} = 50 Hz
Weighting factor	λ_d = 1

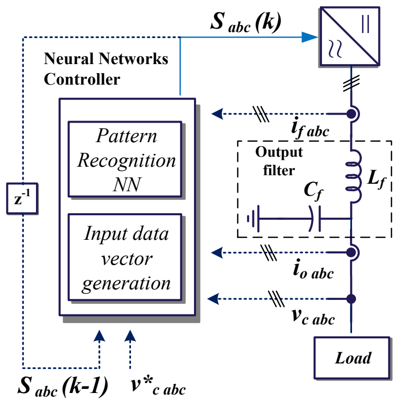


Fig. 1: Simplified scheme of the control algorithm.

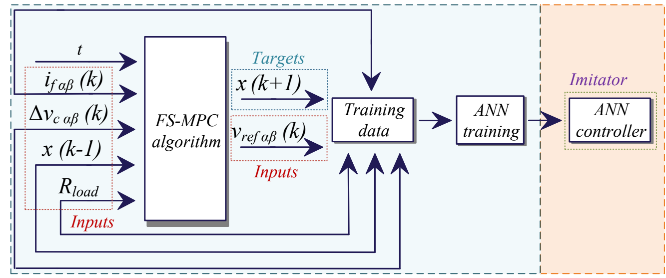


Fig. 2: ANN controller synthesis scheme.

III. Input data generation

The training data has a big influence on the quality of the ANN and special attention is thus given to generation of the training data. In particular, we show that it is not necessary to run a simulation model of the system to generate the data. In contrast, we only use the FS-MPC algorithm function. This allows us to quickly generate vast amounts of data that ensures comprehensive coverage of the controller's action space which is not feasible to do in such short time using the procedure in [14]. When generating the range of the input data it was considered that the data used for training is feasible for the application system and can occur in physical applications. Moreover, in the FS-MPC function used for data generation, delay compensation method presented in [15] was also implemented. Therefore, in the control decision all possible control actions that could have happened in the previous sampling period are considered. If a simulation model was used, there is no certainty that during the simulation run all of these possible combinations were recorded and used for training.

The input matrix of more than 140 million data vectors was generated in only 38 s on 12 paralleled CPUs. As shown in Fig. 2 the FS-MPC algorithm accepts 7 input values: time (t), filter currents ($i_{f\,\alpha}$, $i_{f\,\beta}$), capacitor voltage deviations ($\Delta v_{c\,\alpha}$, $\Delta v_{c\,\beta}$), optimum voltage vector from previous sampling period $x(k-1)$, load resistance (R_{load}) and calculates 3 output values: reference values ($v_{ref\,\alpha}$, $v_{ref\,\beta}$), and future optimum switching combination $x(k+1)$. The reference values $v_{ref\,\alpha\beta}$ are not used in the inputs because the two values are coupled and to keep this dependency it is more convenient to use time vector as an input value. They are calculated using two sine wave equations with a 120° phase shift and the input value of the time vector t.

Following input data was used in the example presented in this paper: time vector $t = [0 : 0.002 : 0.018]$, filter current vectors $i_{f\,\alpha\beta} = [-16 : 3 : 16]$, load vector $R_{load} = [30 : 5 : 60]$, optimum voltage vector from previous sampling period $x_{old} = [1 : 1 : 7]$ and difference between the measured and the reference capacitor voltage vectors $\Delta v_{c\,\alpha\beta} = [-5 : 1 : 5]$. These input vectors form a matrix so all possible combinations are evaluated by the FS-MPC algorithm.

IV. Imitator synthesis

To obtain the imitator, we used a pattern recognition method. The output of the ANN controller, i.e. the $Target$, is a vector with 7 elements. The position of the value 1 in the output vector defines the selected optimum voltage vector. For example, target vector $x(k+1) = [0010000]$ defines the voltage vector V_3. To train the network with 8 input neurons, 15 hidden neurons and 7 output neurons, Neural network pattern recognition app was used. 70% randomly divided input data was used to train the network and 15% for validation and testing. Scaled conjugate gradient (SCG) back-propagation algorithm [29] is used to update the weights and biases in the pattern recognition network according to the scaled conjugate gradient method. SGB back-propagation training algorithm has small memory requirements and it is much faster than standard gradient descent algorithms because it does not require a line search at each iteration. Performance of the training can be observed in the confusion matrix, which is shown in Fig. 3. It can be seen that each row in the matrix corresponds to the predicted class, and the columns correspond the true class, i.e the Target class. The cells in the diagonal of the matrix show how many observations were correctly classified, while all other cells show the incorrect classifications. In the presented example we can see that 97% of the observations were correctly classified. Therefore, we can proceed to the next step and export the trained ANN to Simulink model.

Before the imitator was implemented in the dSPACE real-time platform, a set of simulations was performed to compare the performance with the conventional FS-MPC algorithm from which the imitator was derived. Two performance metrics: THD of the filter capacitor voltage and average switching frequency were used for comparison in Table III. A low THD of the filter capacitor voltage is one of the main requirements for the UPS applications and it is defined in the IEC 62040-3 standard. The results have shown that using the imitator a voltage total harmonic distortion (THD) of 1.42% was obtained, while the conventional algorithm showed a THD of 1.33%. Moreover, average switching frequency was also obtained for the two controllers. The imitator was operating with average frequency of 8 kHz and the conventional algorithm with 8.6 kHz. These performance metrics show that the imitator is capable of imitating the behaviour of the conventional algorithm on a very high level. Simulation models for both the conventional and imitator controller are provided in the supplementary files for horizon prediction 1, 2 and 3. The

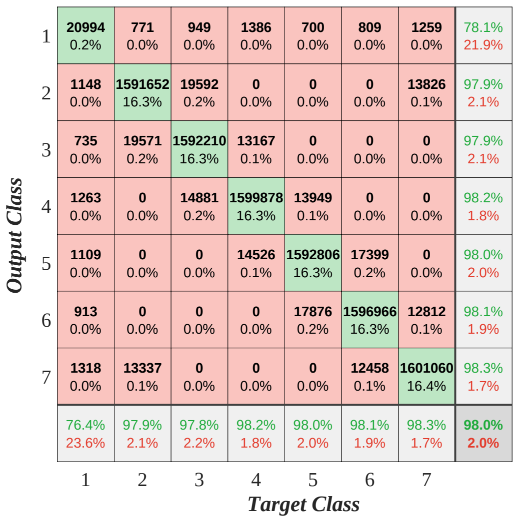


Fig. 3: Confusion matrix of the performed ANN training for 1-step prediction horizon showing the number of correctly (green) and incorrectly(red) classified observations.

imitator controller will also inherit the robustness of the FS-MPC algorithm used for training data generation. This can be verified using the provided simulation models.

A. Application for multi-step prediction horizon algorithms

Increasing the horizon length of FS-MPC can improve the performance of the algorithm [22], [30]. This has been extensively reported in the literature for both simple and complex converter topologies [31]. However, due to the before mentioned problem of high computation, simplifications, linearization of the model and deployment of a search algorithm are necessary [30]. Therefore, by using the same procedure of data generation and NN training it is possible to create an imitator for 2-step horizon and 3-step horizon FS-MPC algorithm. The confusion matrix for 2-step prediction horizon showed misclassification error of 1.7% and for 3-step prediction horizon error of 1.9%. It needs to be mentioned that the classification error depends on division of the data used for training, testing and validation. This is done randomly, therefore, small differences in classification error are expected between training sessions [32].

In Table II influence of the prediction horizon length on the filter capacitor distortion can be observed. The weighting factor λ_d was set to 0, to exclude all effects from this additional term. The results confirm that extension of the horizon can reduce the distortion of the voltage for both the original algorithm and the imitator. Reduction of the sampling frequency will degrade the performance of both controllers, thus by extending the horizon it is possible to improve the reference tracking performance.

B. Application for multilevel and MMC converters

For creating an Imitator controller of a FS-MPC algorithm for multilevel and modular multilevel converters following modifications need to applied to the proposed approach. New

TABLE II: Influence of the prediction horizon length on the capacitor voltage distortion and switching frequency, V_{cref} = 325 V and R_{load} = 60 Ω.

	THD (%)			$f_{sw\ avg}$ (kHz)		
Prediction horizon	1	2	3	1	2	3
FS-MPC controller	1.86	1.37	1.23	7.6	8.4	8.6
Imitator controller	1.97	1.5	1.32	7.2	7.8	8.2

TABLE III: Performance metrics results from simulations and experiments for 1-step prediction horizon, V_{cref} = 325 V and R_{load} = 60 Ω.

	Imitator controller			Conventional controller		
Metrics	THD	f_{sw}	Δv_m	THD	f_{sw}	Δv_m
Sim.	1.42%	8 kHz	8.8%	1.33%	8.6 kHz	7.3%
Exp.	1.76%	7.6 kHz	15.5%	1.69%	8.2 kHz	14%

measurements, which are specific to the application topology (e.g. dc-link capacitor voltages), need to be added to the data generation process. The cost-function of the FS-MPC algorithm that will be used for generating the training data for the NN needs to be modified. An overview of the cost functions for different topologies can be found in [16]. The number of output neurons needs to be increased to the number of possible switching combinations or a new hidden layer, which will be used to code the switching states can be added. There is no straightforward answer which method will perform better. From experience, the method with higher number of output neurons will perform better with the pattern recognition algorithm that was used in the paper. However, maybe a different training algorithm can produce better results for the network with lower number of output neurons. This is still an on-going research topic [32].

V. Experimental validation

For the experimental validation of the imitator and the conventional FS-MPC controller a MicroLabBox DS1202 PowerPC DualCore 2 GHz processor board and DS1302 I/O board from dSpace were used to implement the control algorithm. The experimental set-up is shown in Fig. 4 and the parameters match the simulation parameter values from Table I. The performance metrics in steady state operation for reference voltage V_{cref} = 325 V and R_{load} = 60 Ω were compared for both controllers in Table III. It is observed that the difference in the THD between the two controllers is 0.07% and for the switching frequency the difference is 600 Hz. This confirms very good learning performance of the imitator which was also obtained in the simulations. The transient performance of the controllers was tested on a step load change R_{load} = 60 $\rightarrow$ 30 Ω and the results can be observed in Fig. 5. Maximum voltage error during the load step (Δv_{max}) is shown in Table III. It is evident that the fast transient response of the FS-MPC was also successfully learned by the imitator controller.

VI. Computation burden

The biggest impact of our proposed approach can be seen in the multistep prediction horizon FS-MPC. A comparison

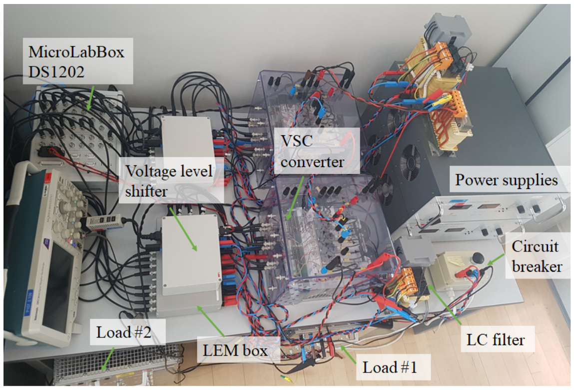
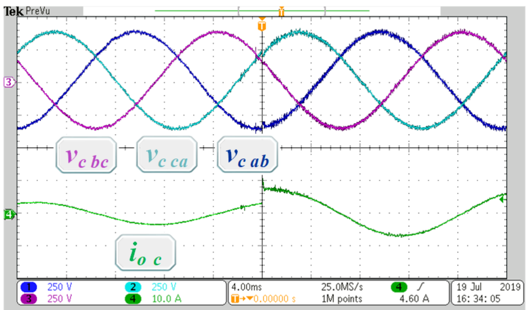
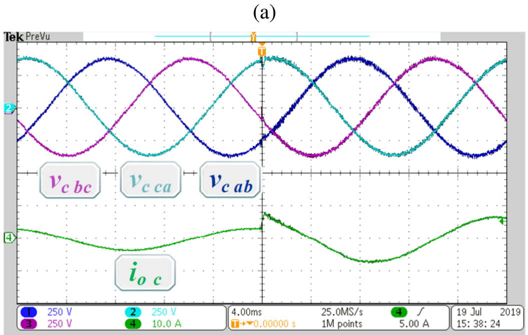


Fig. 4: Two level VSC experimental setup.



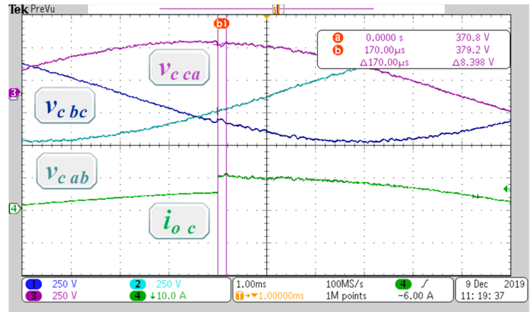
(a)



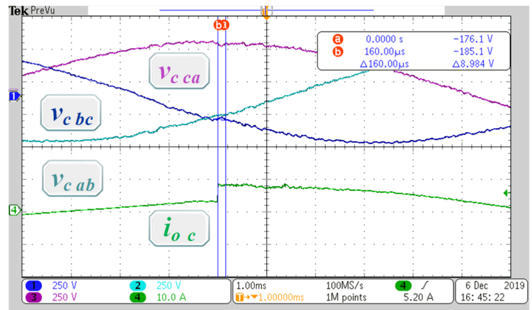
(b)

Fig. 5: Step load transient R_{load} = 60 $\rightarrow$ 30 Ω (1-step prediction horizon): (a) imitator controller, (b) Conventional FS-MPC controller.

of the algorithm execution time and the horizon length for conventional and imitator model was performed in the Fig. 7. The execution time was measured using dSPACE Profiler software, which can analyze timing behavior of applications running on dSPACE platforms. It can be noticed that the execution time for the imitator model is constant as it was not necessary to increase the number of hidden neurons to keep the same percentage of correct identifications in the confusion plot. In sharp contrast, the number of calculation is increasing exponentially for conventional algorithm, which is shown in Fig. 8. This is due to the fact that for 2-step prediction algorithm or 3-step prediction algorithm the number of switching state candidates is also increasing exponentially i.e. for each applied vector in the first step there are 7 vector options for the second step and for each of 49 vectors from



(a)



(b)

Fig. 6: Step load transient R_{load} = 60 $\rightarrow$ 30 Ω: (a) imitator controller (2-step prediction horizon), (b) imitator controller (3-step prediction horizon).

the second step there are again 7 vector options. Thus, for 2-step prediction horizon 49 (7^2) switching combinations are used to calculate the control variables predictions and for 3-step prediction horizon 343 (7^3) switching combinations are possible. It needs to be noted that execution time under 20 μs is not possible without overruns for 3-step prediction horizon (see Fig. 7). In this letter the algorithm was implemented on a DSP platform, a further decrease of the turn around time can be expected from a implementation on a FPGA platform. In [33] it was demonstrated that by sharing the resources, the execution time of a FS-MPC algorithm can be decreased. However even so, the FPGA's still does not have enough registers to implement the FS-MPC algorithm for higher horizons without using the search algorithms to reduce the number of switching candidates. Therefore, we believe that our proposed approach of imitating the multi-step horizon controller with an NN structure could achieve this goal. The NN structure has a simple implementation on a FPGA platform and the calculations can be parallelized to reduce the execution time.

VII. Conclusion

The paper presented a new controller synthesis approach for power electronic converters. An ANN based imitator model of FS-MPC was trained using pattern recognition algorithm and successfully validated in experiments with a very good accuracy. Therefore, using this approach imitator controller can inherit the good performance of the FS-MPC algorithm with reduced computational burden. Findings in this paper open many new possibilities for future development of the

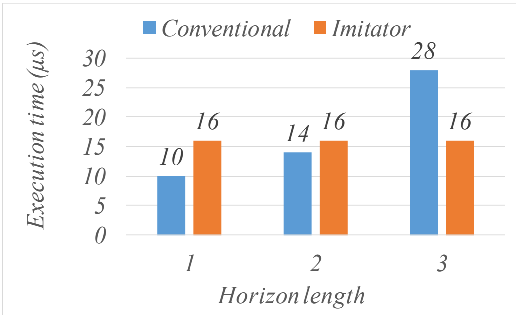


Fig. 7: Comparison of the algorithm execution time and the horizon length.

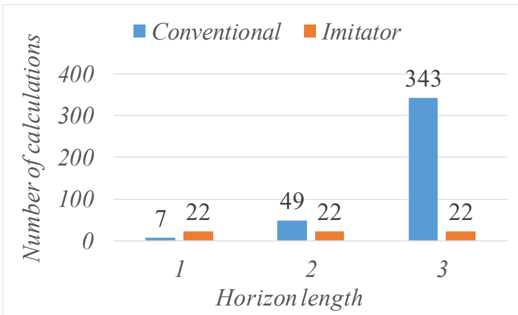


Fig. 8: Comparison of the number of calculations and the horizon length.

presented approach. From one point it gives an opportunity for implementation of computational heavy predictive algorithms with multi-step predictions or algorithms for multi-level or modular multi-level topologies. Moreover, there is a possibility that the imitator could also be tuned online using the new measurements obtained during the operation. This would of course improve the performance of the controller even more. Therefore, future development of this approach could be of big interest also for the industry.

Acknowledgment

Author Tomislav Dragicevic would like to thank Mr. Ihab S. Mohamed for valuable initial discussions that preceded the preparation of this paper.

References

[1] F. Harashima, Y. Demizu, S. Kondo, and H. Hashimoto, “Application of neutral networks to power converter control,” in *Conf. Record of the IEEE Ind. Appl. Society Annual Meeting,*, DOI 10.1109/IAS.1989.96777, pp. 1086–1091 vol.1, Oct. 1989.

[2] B. R. Lin and R. G. Hoft, “Power electronics inverter control with neural networks,” in *Proc. of 8th Annual Applied Power Electron. Conf. and Expo.*,, DOI 10.1109/APEC.1993.290640, pp. 128–134, Mar. 1993.

[3] K. T. Chau and C. C. Chan, “Real-time implementation of an on-line trained neural network controller for power electronics converters,” in *Proc. of 1994 Power Electronics Specialist Conf. - PESC’94*, vol. 1, DOI 10.1109/PESC.1994.349714, pp. 321–327 vol.1, Jun. 1994.

[4] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, “Training recurrent neural networks with the levenberg marquardt algorithm for optimal control of a grid-connected converter,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 26, DOI 10.1109/TNNLS.2014.2361267, no. 9, pp. 1900–1912, Sep. 2015.

[5] Y. Sun, S. Li, B. Lin, X. Fu, M. Ramezani, and I. Jaithwa, “Artificial neural network for control and grid integration of residential solar photovoltaic systems,” *IEEE Trans. Sustainable Energy*, vol. 8, DOI 10.1109/TSTE.2017.2691669, no. 4, pp. 1484–1495, Oct. 2017.

[6] D. Wang and *et al.*, “A deep neural network based predictive control strategy for high frequency multilevel converters,” in *2018 IEEE Energy Conversion Congress and Expo. (ECCE)*, DOI 10.1109/ECCE.2018.8558293, pp. 2988–2992, Sep. 2018.

[7] Y. Pan and J. Wang, “Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks,” *IEEE Trans. Ind. Electron.*, vol. 59, DOI 10.1109/TIE.2011.2169636, no. 8, pp. 3089–3101, Aug. 2012.

[8] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, “Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems,” *Energy and Buildings*, vol. 141, DOI 10.1016/j.enbuild.2017.02.012, pp. 96 – 113, 2017.

[9] S. Piche, B. Sayyar-Rodsari, D. Johnson, and M. Gerules, “Nonlinear model predictive control using neural networks,” *IEEE Control Systems Magazine*, vol. 20, DOI 10.1109/37.845038, no. 3, pp. 53–62, Jun. 2000.

[10] S. Stevsic, T. Nageli, J. Alonso-Mora, and O. Hilliges, “Sample efficient learning of path following and obstacle avoidance behavior for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 3, DOI 10.1109/LRA.2018.2856922, no. 4, pp. 3852–3859, Oct. 2018.

[11] 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 Conf. and Expo (ITEC)*, pp. 84–89, Jun. 2017.

[12] M. Hertneck, J. Kohler, S. Trimpe, and F. Allgower, “Learning an approximate model predictive controller with guarantees,” *IEEE Control Systems Letters*, vol. 2, DOI 10.1109/LCSYS.2018.2843682, no. 3, pp. 543–548, Jul. 2018.

[13] D. Georges, “A simple machine learning technique for model predictive control,” in *2019 27th Mediterranean Conf. on Control and Automation (MED)*, DOI 10.1109/MED.2019.8798512, pp. 69–74, Jul. 2019.

[14] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicevic, and A. A. Z. Diab, “A neural-network-based model predictive control of three-phase inverter with an output LC filter,” *IEEE Access*, vol. 7, DOI 10.1109/ACCESS.2019.2938220, pp. 124 737–124 749, 2019.

[15] J. Rodriguez and P. Cortes, *Predictive Control of Power Converters and Electrical Drives*, ser. Wiley - IEEE. Wiley, 2012.

[16] 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, DOI 10.1109/TIE.2016.2625238, no. 2, pp. 935–947, Feb. 2017.

[17] S. R. Mohapatra and V. Agarwal, “Model predictive controller with reduced complexity for grid-tied multilevel inverters,” *IEEE Trans. Ind. Electron.*, vol. 66, DOI 10.1109/TIE.2018.2866115, no. 11, pp. 8851–8855, Nov. 2019.

[18] A. Dekka, B. Wu, V. Yaramasu, and N. R. Zargari, “Dual-stage model predictive control with improved harmonic performance for modular multilevel converter,” *IEEE Trans. Ind. Electron.*, vol. 63, DOI 10.1109/TIE.2016.2571259, no. 10, pp. 6010–6019, Oct. 2016.

[19] M. Siami, D. Arab Khaburi, and J. Rodriguez, “Simplified finite control set-model predictive control for matrix converter-fed PMSM drives,” *IEEE Trans. Power Electron.*, vol. 33, DOI 10.1109/TPEL.2017.2696902, no. 3, pp. 2438–2446, Mar. 2018.

[20] B. Gutierrez and S. Kwak, “Model predictive control method with preselected control options for reduced computational complexity in modular multilevel converters (MMCs),” in *2018 20th European Conf. on Power Electron. and Appl. (EPE’18 ECCE Europe)*, pp. P.1–P.8, Sep. 2018.

[21] T. Geyer, *Model Predictive Control of High Power Converters and Industrial Drives*. Wiley, 2016.

[22] E. Liegmann, P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, “Long-horizon direct model predictive control with active balancing of the neutral point potential,” in *2017 IEEE International Symp. on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, DOI 10.1109/PRECEDE.2017.8071274, pp. 89–94, Sep. 2017.

[23] P. Cortes, G. Ortiz, J. I. Yuz, J. Rodriguez, S. Vazquez, and L. G. Franquelo, “Model predictive control of an inverter with output LC filter for UPS applications,” *IEEE Trans. Ind. Electron.*, vol. 56, DOI 10.1109/TIE.2009.2015750, no. 6, pp. 1875–1883, Jun. 2009.

[24] T. Dragicevic, “Model predictive control of power converters for robust and fast operation of AC microgrids,” *IEEE Trans. Power Electron.*, vol. 33, DOI 10.1109/TPEL.2017.2744986, no. 7, pp. 6304–6317, Jul. 2018.

[25] P. Cortes and *et al.*, “Guidelines for weighting factors design in model predictive control of power converters and drives,” in *2009 IEEE Int. Conf. on Ind. Technology*, pp. 1–7, Feb. 2009.

[26] P. R. U. Guazzelli, W. C. de Andrade Pereira, C. M. R. de Oliveira, A. G. de Castro, and M. L. de Aguiar, “Weighting factors optimiza-

tion of predictive torque control of induction motor by multiobjective genetic algorithm," *IEEE Trans. Power Electron.*, vol. 34, DOI 10.1109/TPEL.2018.2834304, no. 7, pp. 6628–6638, Jul. 2019.

[27] 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.*, vol. 66, DOI 10.1109/TIE.2018.2875660, no. 11, pp. 8870–8880, Nov. 2019.

[28] "IEC standard for uninterruptible power systems (UPS) - part 3," *IEC 62040-3:2011*, no. 2, pp. 1–214, Mar. 2011.

[29] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, DOI 10.1016/S0893-6080(05)80056-5, no. 4, pp. 525 – 533, 1993.

[30] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 30, DOI 10.1109/TPEL.2014.2316173, no. 3, pp. 1633–1644, Mar. 2015.

[31] C. Bordons and C. Montero, "Basic principles of mpc for power converters: Bridging the gap between theory and practice," *IEEE Ind. Electron. Magazine*, vol. 9, DOI 10.1109/MIE.2014.2356600, no. 3, pp. 31–43, Sep. 2015.

[32] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[33] S. Wendel, A. Dietz, and R. Kennel, "FPGA based finite-set model predictive current control for small PMSM drives with efficient resource streaming," in *2017 IEEE Int. Symp. on Predictive Control of El. Drives and Power Electron. (PRECEDE)*, DOI 10.1109/PRECEDE.2017.8071270, pp. 66–71, Sep. 2017.