A Self-learning Scheme to Detect and Mitigate the Impact of Model Parameters Imperfection in Predictive Controlled Grid-tied Inverter

Matthew Baker, *Student Member*, *IEEE*, Hassan Althuwaini, *Student Member*, *IEEE,* and
Mohammad B. Shadmand, *Senior Member*, *IEEE*

Intelligent Power Electronics at Grid Edge (IPEG) Research Laboratory
Electrical and Computer Engineering Department, University of Illinois at Chicago, USA
mbaker36@uic.edu; halthu2@uic.edu; shadmand@uic.edu

***Abstract*—This paper presents a neural-network based self-learning mechanism for improving the performance of model predictive control (MPC). Model parameters mismatch in MPC can occur due to manufacturing variance, temperature variance, component aging, loading condition or other sources. Model uncertainties decreases the overall efficiency of the MPC leading to non-optimal switching sequence generation. To mitigate mismatch, this paper proposes an artificial intelligence (AI) scheme to provide model parameter healing in real-time. Two AI approaches are evaluated. The first approach is a classification two-steps network, and the second approach is a model adaptation network. Fine tree and feedforward neural networks are trained to implement these layers. The proposed neural network schemes are verified to correct for mismatch, then compared to each other to find the optimal solution for a grid-interactive inverter. Several case studies provided to validate the theoretical expectations.**

***Index Terms*— model predictive control, machine learning, artificial intelligence, smart inverter, self-healing control.**

I. Introduction

Advancements in computational processing power in the last decade enabled model-based controllers with online optimization such as model predictive control (MPC). Recent studies in academia and industry demonstrates superiority of MPC over classical control such as fast dynamic response, elimination of tuning effort and the simple inclusion of constrains [1, 2]. For instance, in comparison to proportional-resonant control (PR) and other classical control methods for power electronic interfaces, MPC demonstrates superior performance with minimal tuning [3-6]. Among different MPC categories, finite-set MPC (FS-MPC) features such as online optimization can improve the dynamic performance and minimize tracking error which can eventually improve the resiliency of power electronic systems [7, 8].

The primary feature of MPC is that it uses a model of the system to predict the future behavior of the control objectives for each potential switching state of a power converter. For proper operation and prediction of control variables, MPC requires accurate values of system parameters, such as filter inductance. Additionally, for a grid-interactive inverter application, other external variables such as ambient temperature, aging, and magnetic saturation might create changes in the model impedance parameters, resulting in a mismatch and causing an imperfect prediction of control variables for MPC cost function that determines the optimal switching sequences [9, 10]. The impact of model parameter mismatch in power electronics applications has been empirically investigated in the literature by looking at inverter behavior under various condition and assessing the power quality [11]. In [11], an auto-tune heuristic algorithm is implemented used to fine-tune the parameters of the model, [12] has implemented robust MPC algorithm with parallel compensation terms against multi-parameter mismatches. An advanced state observer is developed in [13] to estimate interference caused by model parameter mismatches, and mitigated the impact with a feedforward loop. In [14], the effects of model parameter uncertainty on prediction error have been investigated, but no solutions for mitigating the model parameter mismatch problem have been proposed. The authors in [15] proposed a predictive controller based on parameter estimation instead of a pre-defined model, and [16] considers an algorithm for online parameter identification of inductance and flux linkage of motor. These approaches improve the MPC performance, but the main drawbacks are accuracy of detecting the mismatch and needed correction.

This paper proposes a data-driven approach coupled with MPC for realization of a highly accurate model parameter mismatch detection and self-healing process without requirement of an additional control loop or complex computation. Two artificial intelligence (AI) methods are investigated for tackling the stated problem in grid-interactive inverter applications. The first approach is a one-network scheme which utilizes a feedforward neural network as a model adaptation network to find the extent of mismatch. The second approach is a two-networks scheme, which consists of a classification network, followed by the model adaptation network. Both feedforward network and fine tree classification network topologies are investigated in this paper. When it detects mismatch, the model adaptation network is executed. The model adaptation network is integrated with a neural network approach through a healing factor, L_{Δ}, for adapting the model during the optimization. The objective of the healing factor L_{Δ} is to mitigate the mismatch between the model value and the actual filter inductance, which is an uncertainty in model due to a variety of external agents including ambient temperature, component aging, and magnetic saturation. The

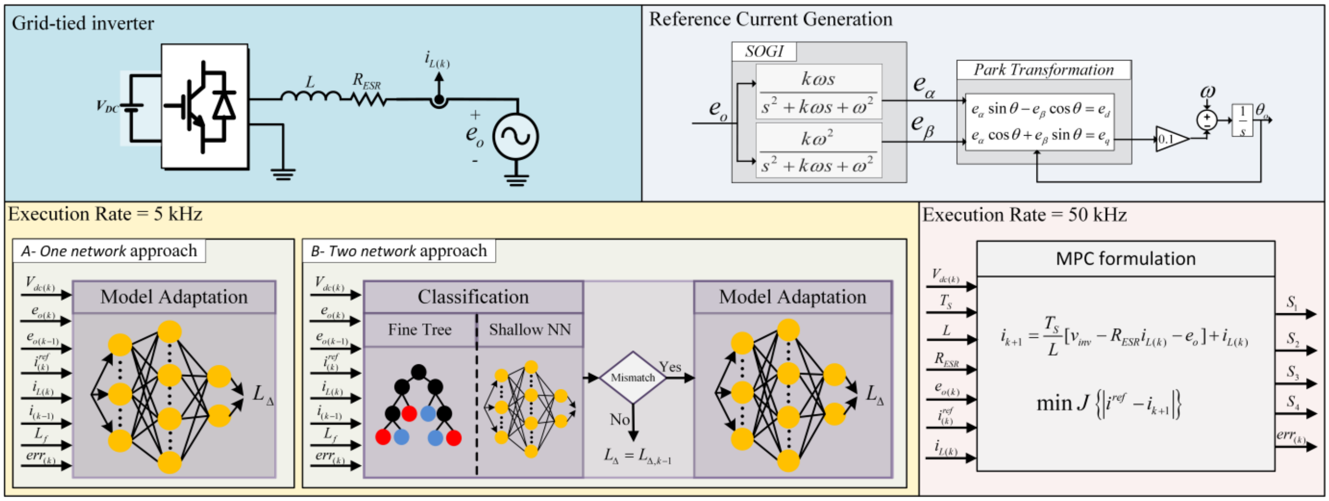


Fig. 1. Proposed self-learning MPC for grid-interactive inverters with disturbance mitigation capability. Two different network approaches are studied in this paper: A shows the one-network network which utilizes the model adaptation network every 5 kHz, thus adjusting L_Δ regardless of potential mismatch. The second approach B is the two-network network which only runs the model adaptation network when the classification network indicates there is mismatch. Therefore, only the classification network is guaranteed to operate every 5 kHz.

proposed control mitigates the impact of model parameter mismatch on power quality and tracking error.

The remainder of this paper is structured as follows; Section II covers the mathematical derivation of MPC as a groundwork for the proposed control scheme, Section III details the network training and generation, Section IV covers case studies demonstrating the impact of the proposed approach for improving the MPC performance, and Section V concludes the paper.

II. Ground Work of Model Predictive Control

An overview of the illustrated control is presented in Fig. 1. The inverter is interfaced with an inductor L. The grid angle is detected, and the reference is split to the $\alpha\beta$-framework using a phase locked loop (PLL) with a second order generalized integrator (SOGI)-based orthogonal signal generator (OSG). A more detailed explanation of the MPC derivation is found in [8, 17], this section provides a brief overview of MPC formulation. The reference current is constructed in the dq framework as,

$$\begin{aligned} i^{ref}_{d,k} &= \frac{2(P^{ref}_k e_{d,k} + Q^{ref}_k e_{q,k})}{{e_{d,k}}^2 + {e_{q,k}}^2} \\ i^{ref}_{q,k} &= \frac{2(P^{ref}_k e_{q,k} - Q^{ref}_k e_{d,k})}{{e_{d,k}}^2 + {e_{q,k}}^2} \\ i^{ref}_k &= i^{ref}_{d,k}\sin(\theta_k) + i^{ref}_{q,k}\cos(\theta_k) \end{aligned} \tag{1}$$

where P^{ref}_k and Q^{ref}_k are active and reactive power, $e_{d,k}$ and $e_{q,k}$ are the grid voltage, and i_d and i_q are the decoupled current components in the rotating frame. The dynamic model of the L filter is given by,

$$\frac{di}{dt} = \frac{1}{L}(v_{inv} - R_{ESR} i_{L(k)} - e_o) \tag{2}$$

where v_{inv} is the bridge voltage of the inverter and e_o is the grid voltage, L is the learned filter inductance which will be defined later and R_{ESR} is the equivalent series resistance. The Euler forward method is applied to the discretize (2) and predict the current one-step ahead in horizon of time as following,

$$i_{k+1} = \frac{T_S}{L}[v_{inv} - R_{ESR} i_{L(k)} - e_o] + i_{L(k)} \tag{3}$$

where i_{k+1} is the one step ahead prediction of the current, and T_s is the sampling time. To detect mismatch, the value of L used in (3) is defined as

$$L = L_f + L_\Delta \tag{4}$$

where L_f is the modeled inductance value, and L_Δ is the predicted mismatch between the modeled value and actual values. $L_f$is immutable, while L_Δ is the healing factor which is adaptively changed based on the neural network approach detailed in Section III.

Finally, a cost function is constructed to determine the optimal switching sequence and tracking the desired grid current,

$$J = |i^{ref}_k - i_{k+1}| \tag{5}$$

III. Network Training and Generation

A. Training Data Collection

Two neural network approaches are used and compared in this paper. The first approach is a classification network; the second approach is a model adaptation network. The classification network determines if there is mismatch in the filter. When using the classification network, if model mismatch is detected, the output of the classification network activates the model adaptation network to adjust L_Δ. Thus, when using the classification network, a two-network approach is used to adjust mismatch. Conversely, the other neural network proposed consists only of the model adaptation network. The model adaptation network is a neural network which determines the healing factor for the system. Thus, to adjust the network this model adaptation network is mandatory. This paper determines whether the classification network provides utility. It will benefit the system if it prevents the model adaptation network from executing when mismatch is negligible. If not, only the model adaptation network is needed.

To initiate the neural network training process, data must be collected to train the system. MATLAB/Simulink is used for all data collection and simulations in this training process. The data is collected by executing the system of Fig. 1 at several values of L. By varying L, the training network can adapt to whichever filter size is needed by the system designer. Model mismatch is intentionally created in the training data set by iteratively changing the percent error mismatch needed in L. This intentional mismatch is pivotal as it allows the neural network to have enough data for accurate determination of a change in the filter inductance. For the iterative process, L is varied from 0 to 5 mH in 0.5 mH increments. This is equivalent to L_f in (4). Then, mismatch is created by altering the actual filter inductance by a certain percentage of the L value. This value is again iteratively changed from 50% to 150% of the initial value in 10% increments. This is equivalent to L_Δ in (4). The system is simulated, and data is sampled at rate of 5kHz to generate the training data for the system. The system dc-bus voltage is V_{dc}=200V, active and reactive power references are P_k^{ref}= 3kW, Q_k^{ref} =0 var, and the grid voltage is 120 $\mathrm{V_{rms.}}$ At each sampling instant k, the following variables are collected to fill out the array INP,

$$INP = \begin{bmatrix} err_{(k)}, V_{dc(k)}, e_{o(k)}, i_\alpha^{ref} \\ i_{L(k)}, L_f, e_{o(k-1)}, i_{L(k-1)} \end{bmatrix} \tag{6}$$

where err(k) is the mean square current tracking error, V_{dc}(k) is the dc-link voltage, e_o(k) is the voltage at the point of common coupling, i_α^{ref} (k) is the reference current , i_L(k) is the measured inductor current, err(k) is the mean square current tracking error between i_α^{ref} (k) and i_L(k), and L_f (k) is the reference model inductance. e_o(k-1) and i_L(k-1) are the previous voltage and current measurements, respectfully. The result of each data sample is a 1x8 array, the neural network training input. The classification training data will have an output value of either 0 or 1 indicating a logic LOW or HIGH. A HIGH value denotes there is no mismatch, and the system is operating properly, while and LOW indicates there is mismatch to be corrected. The model adaptation training data likewise has a single output value per sample indicating the value of L_Δ. These data are divided into training, verification, and testing data with ratio of 70%, 15%, and 15% respectfully.

Two neural network approaches are trained to determine which is more accurate for classification data. One deep learning approach, the fine tree classification network, and another a shallow learning approach, a feedforward artificial neural network.

TABLE. I: FINE TREE CLASSIFICATION ACCURACY

Maximum Number of Splits	Accuracy
8	59.7%
16	61.2%
32	63.0%
64	64.6%

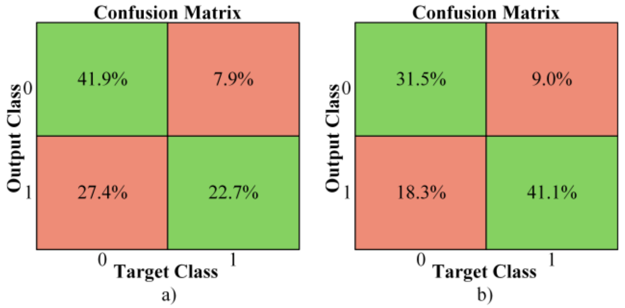


Fig. 2 Best confusion matrix of the (a) fine tree and (b) shallow neural network classification networks. The "1" class indicates no mismatch and "0" indicates mismatch. The green squares indicated the correct classifications, while red squares are incorrect classifications. By adding the correct classifications, the total accuracy is recorded as 64.6% in (a), 72.6% in (b)

B. Classification Network Training

The fine tree classification network creates a decision tree which uses weighted values to determine the next branch for each input [18]. Classification is determined when an input reaches the terminal node of the tree, each of which is assigned to a specific class. These weights are determined by the neural network training process. The fine tree classification network has an advantage as a tree's asymmetry can lead to quick classification of well-known decision paths, and only use longer paths for features which require the extra computation. The fine tree approach is compared to the shallow neural network approach. A shallow neural network consists of an input layer, several intermediate hidden layers, and an output layer. Simplified diagrams of a fine tree classification network and shallow neural network are illustrated in Fig. 1.

In the fine tree classification network, the relevant hyperparameter is the number of decision nodes given in the tree. The MATLAB Deep Learning Toolbox is used to determine the weights of each split and the only variable hyperparameter is the number of decision nodes. For the shallow neural network, the hyperparameters adjusted are the number of hidden layers and the number of neurons per layer. These hyperparameters were determined heuristically to see the effect on accuracy, and the MATLAB 'train()' command is used to train the neural network. The hyperparameters chosen and the accuracy of each result are in Table I for the fine tree classification and Table II for the shallow neural network. The tree classification accuracy struggles to achieve accuracy beyond 64.6% as shown in the confusion matrix in Fig. 2a. Only marginal improvements are observed as the number of splits increases exponentially. On the other hand, the shallow neural network classification can outperform the fine tree classification across almost all hyperparameters chosen. The neural network classification accuracy tends to increase as the number of neurons increases as well as when the number of hidden layers increases. Thus, the best classification network scheme is the classification network with three hidden layers consisting of 16 neurons in layer one, 32 neurons in layer two, and 16 neurons in layer three. These results in a classification accuracy of 72.63% and yields the confusion matrix shown in Fig. 2b.

C. *Model Adaptation Network Training*

As the shallow neural network consistently performed better than the fine tree network in the training process for the classification network, only the shallow neural network is considered for the model adaptation network. The testing data used matches the training data acquired from Section III-A. However, instead of a Boolean output, the neural network outputs the estimated value for the healing factor L_Δ. Thus, instead of determining a new value for L, the model mismatch detection network is designed to simply add or subtract L_Δ from L_f and use it in determining the cost function (5). The mismatch is corrected using this method without complex algorithms and computationally heavy heuristic approaches. This allows the proposed mismatched detection method to be used in any MPC controller with minimal invasion.

The 'train()' command is used in MATLAB to determine the optimal hyperparameters for the shallow neural network. The input data for each instant is the array *INP* and the output datum is L_Δ. The hyperparameters considered for this training network are the same as the shallow neural network in the Section III-B. The maximum number of neurons considered, 64, is chosen to minimize the total training time. More neurons could be considered for training, should designers substantially increase training time or processing power. Additionally, the number of layers and number of neurons per layer could be fine tuned to include more configurations. However, expanding the number of layouts increases the total training time. Hyperparameter optimization methods [19] may be used to find maximally optimal hyperparameters efficiently, but determining them though these methods is beyond the scope of this paper.

Defining accuracy for the model adaptation network is more complicated than for the classification network. This is because each classification prediction is either correct or incorrect, as the data is Boolean. However, for the model adaptation network, each prediction will have some error compared to the actual value. The determination of the most accurate model adaptation network is therefore determined as the number of predictions within a determined accuracy range. This is visualized by plotting the histogram of the predictions as a percentage error of the actual value, as it is demonstrated in Fig. 3. The predetermined accuracy range is the tolerance band between the black lines and the number of values in this band are considered accurate. So, whichever neural network has the most samples between these values is considered the most accurate. For this work, 2% error above or below the actual inductance value is considered acceptable. So, the neural networks are trained based on the hyperparameters listed in Table III and the accuracy shown is the percentage of the testing data with 2% or lower error. From these results, it is again observed that the 3-layers, 16, 32, 16 neuron configuration yields the most accurate neural network with 85.11% of testing data within the tolerance band. Fig. 4a shows the histogram of this neural network, displaying the error bins between -5% and 5% testing error. The mean of the data set is 0.444, while the standard deviation is unexpectedly high at 8.3. The high standard deviation appears to occur because of a small number of incredibly high outliers in the prediction. Fig. 4b shows the histogram of the entire span of the prediction network, vertically scaled to only show the first few instances. The most error-prone bins go as low as a -80% error to a 100% error. Few predictions are highly inaccurate as demonstrated by the high number of bins within the tolerance bands. These outliers cause the standard deviation to exceed 2%, where it would be expected to be since more than 68% of data values are within this variance band. So, while it cannot be assumed this data set follows the normal distribution, the high number of values within the tolerance band indicate practical applications of the neural network are achievable.

IV. Simulation And Results

A. *One-network and Two-networkComparison*

Practical applications of the neural network mismatch approach are examined through two case studies. The first case

TABLE. II: NEURAL NETWORK CLASSIFICATION ACCURACY

Hidden Layers	Neuron Configuration	Accuracy
1	8	60.52%
1	16	61.11%
1	32	62.55%
1	64	59.56%
2	6, 2	58.46%
2	12, 4	60.45%
2	24, 8	62.75%
2	48, 16	62.79%
3	2, 4, 2	59.85%
3	4, 8, 4	60.87%
3	8, 16, 8	64.13%
3	16, 32, 16	72.63%

TABLE. III: MODEL ADAPTATION NETWORK ACCURACY

Hidden Layers	Neuron Configuration	Accuracy
1	8	7.60%
1	16	14.27%
1	32	12.97%
1	64	11.46%
2	6, 2	7.92%
2	12, 4	28.70%
2	24, 8	1.205
2	48, 16	2.77%
3	2, 4, 2	8.23%
3	4, 8, 4	57.23%
3	8, 16, 8	80.77%
3	16, 32, 16	85.11%

study will only consider the model adaptation neural network, while in the second, the classification network is used in conjunction with the model adaptation network. In the single neural network approach, the model adaptation network is executed at a set rate of 5 kHz. The MPC operates at 50 kHz. Thus, the entire neural network is executed at 5 kHz regardless of parameter mismatch. To filter out the high outliers mentioned in Section III, a moving average block is used to prevent false increases of the healing factor. In this case study, the proposed approach is tested first to a single step change in L. In this instance L increases from 2mH to 3.5 mH, so the expected L_Δ increases from 0 to 1.5 mH at time 0.5s. The results of the case study are in Fig. 5. This case study is repeated for the two-network approach and the response is captured in Fig. 6. Like the one-network approach, this approach is also able to adjust the value of L_Δ with a near identical response and accuracy. When combining the two-network, near identical results are realized from the system response.

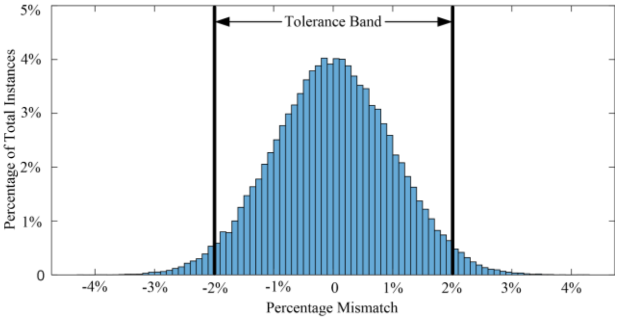
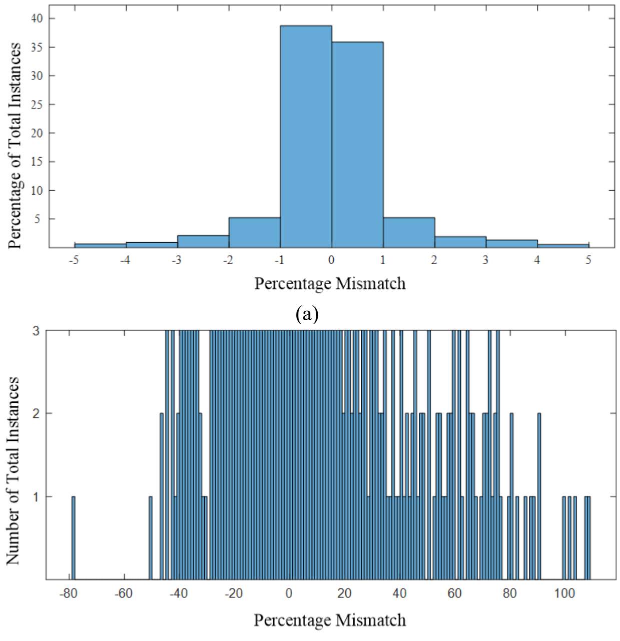


Fig. 3 Example histogram to demonstrate the tolerance band which is acceptable in the model adaptation neural network. The accuracy of the neural network is defined as the total percentage of instances which fall within the tolerance band which is ±2%



(a)

(b)

Fig. 4 Histogram of best Model Adaptation Network (a) demonstrating the error bins from -5% to 5% error, which shows most of the prediction mismatch and that 85.1% of results are within the 2% tolerance band (b) the entire span of the histogram showing a small percentages of very large prediction errors skew the standard deviation, however they do not affect the practical applications of the neural network.

Crucially, the similarity of the two responses indicates the classification network is not adding sufficient utility to the network. To examine this cause, the output of the classification network during the step change is recorded in Fig. 7. The classification is predicting false negatives and positives which result in the model updating both when the mismatch is zero and not triggering when there is mismatch. Because of classification inaccuracy, the system response updates the healing factor almost every sampling instance. Thus, it is essentially the single network approach but with an increased computational burden.

To verify if there is substantial difference between the approaches in other operating conditions, the mismatch study is expanded to test the entire range of mismatch percentage from the training set. Thus, an initial L_f value of 2 mH, the actual L is varied from 1mH to 3mH in 0.2mH intervals. Each mismatch scenario is executed for duration of 1s to allow sufficient time for L_Δ to be determined and an average error calculated. The resulting accuracy is the mean percentage error between the estimated L and the actual value L in that scenario. This process is executed twice, with the one-network and again for the two-network configuration. The results are recorded in Table IV. The two-network response is more accurate in six scenarios, while the one-network approach is best in five scenarios. There are some situations where the two-network approach is superior, particularly for lower values of L_Δ. However, at an accuracy rate of 72%, the model adaptation network is still running most of the time, meaning a higher computational burden presents for only slight accuracy gains in a small range of operating conditions. Therefore, with the reported accuracy of the classification network, it is suggested the best approach is to use only the model adaptation network. A two-network scheme needs to increase classification accuracy before practical implementation.

B. *Power Quality Analysis and Proposed Control Performance*

The impact of the proposed self-learning MPC on power quality is evaluated next. In this case study, the one-network

TABLE. IV: MODEL ADAPTATION NETWORK ACCURACY

L_f (mH)	One-network Error	Two-network Error
1.0	12.60%	10.70%
1.2	6.25%	5.17%
1.4	5.57%	5.50%
1.6	3.69%	2.81%
1.8	1.56%	1.78%
2.0	1.45%	1.60%
2.2	0.36%	0.32%
2.4	0.42%	0.46%
2.6	-0.19%	-0.27%
2.8	-0.25%	-0.21%
3.0	-0.57%	-0.60%

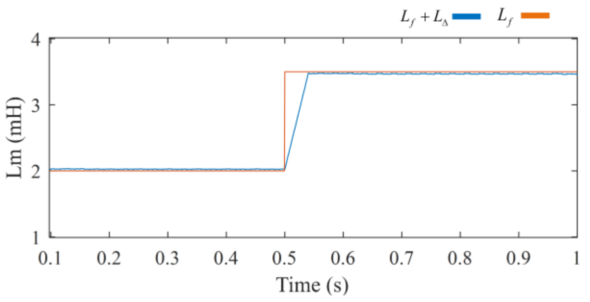


Fig. 5 The response of the one-network prediction where L increases 1.5 mH at time 0.5 seconds. Mismatch error is 0.60%

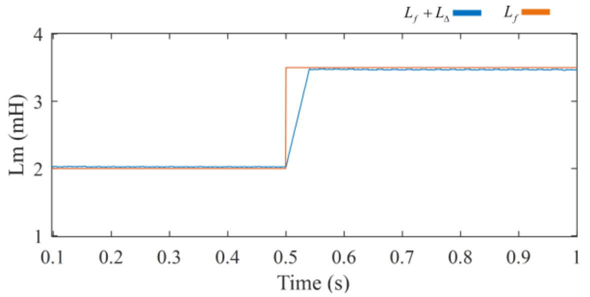


Fig. 6 The response of the two-network prediction where L increases 1.5 mH at time 0.5 seconds. Mismatch error is 0.57%

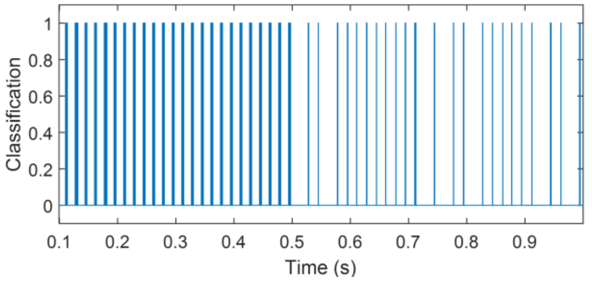
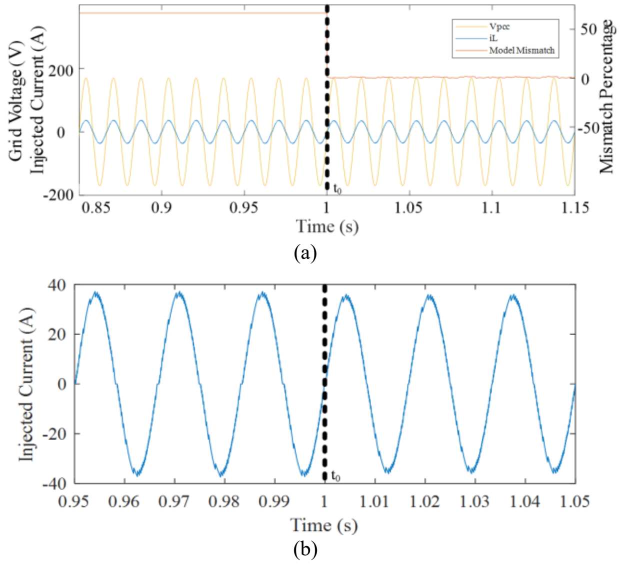


Fig. 7 Output of the classification network during the two-network scheme in Fig. 6. The inability of the mismatch scheme to have accuracy higher than the model adaptation network implies that it operates near identically to the one-network scheme.

approach is used to show the system adaptation from an inductance mismatch where L_f= 2mH and L= 6mH. The model adaptation model is initialized at t=1s. In Fig. 8a, the resulting waveform is shown when the adaptation is made. The predicted inductance increases to 5.981mH, a prediction error of -0.32%. Fig.8b provides a close view of the injected current immediately before and after mismatch correction. Fig. 9a provides the frequency spectrum before the correction, and Fig. 9b afterwards. The mismatch mitigation provided by the neural network results in a THD improvement from 4.09% to 2.58%, verifying improved conditions when implementing the proposed scheme.

V. Conclusion

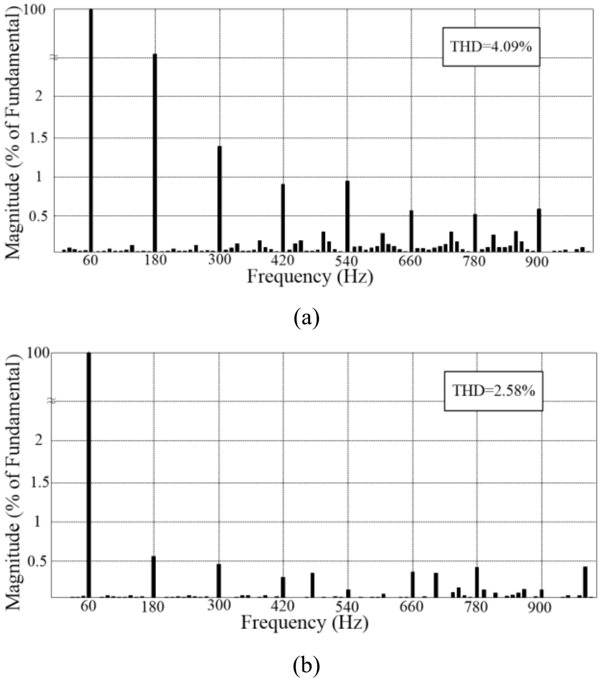
This paper presented a self-learning MPC to tackle the model parameters imperfection and their impacts on the controller performance, thereby maximizing the superiority of MPC in comparison to classical control schemes for power electronics interfaces. The proposed model parameter mismatch detection method can accurately predict the value of a defined healing factor L_Δ and utilizes it in model predictive control. This novel mismatch approach uses neural networks trained to detect changes to the system parameters which indicate filter mismatch. Two neural network classification



(a)

(b)

Fig. 8 (a) Waveforms of grid-tied inverter before and after the one-network model adaptation network is implemented. At t_o=1s the neural network adaptation network updates the model inductance which results in the mismatch percentage dropping from 66% to 0.32% (b) close-up of current waveform showing the improved waveform after t_0



(a)

(b)

Fig. 9 Frequency analysis of the injected currant (a) before the mismatch detection and healing activation (conventional MPC) (b) after detection and self-healing triggered (proposed self-learning MPC). THD is reduced from 4.09% to 2.58%

techniques, fine tree classification and shallow neural network, are used to determine if there is an error and another feedforward network adapts the model when this classification occurs. Both approaches are evaluated for the application in hand. While both can adapt to the change in inductance, it is recommended that only the model adaptation network is used unless classification accuracy can match or exceed the accuracy of the model adaptation network, as the large number of false positives lead to the classification network triggering the model adaptation network most sampling instances. The system is verified with a frequency spectrum analysis which demonstrates the mismatch correction improves THD of injected grid current from 4.09% to 2.58% when MPC experiences uncertainties in model parameters.

Acknowledgment

This work was supported by the U.S. National Science Foundation under Grant ECCS-2114442. The statements made herein are solely the responsibility of the author.

References

[1] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive Control in Power Electronics and Drives," *IEEE Transactions on Industrial Electronics,* vol. 55, no. 12, pp. 4312-4324, 2008, doi: 10.1109/TIE.2008.2007480.

[2] T. Geyer, *Model predictive control of high power converters and industrial drives*. John Wiley & Sons, 2016.

[3] M. Easley, M. B. Shadmand, and H. Abu-Rub, "Hierarchical Model Predictive Control of Grid-Connected Cascaded Multilevel Inverter," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 9, no. 3, pp. 3137-3149, 2021, doi: 10.1109/JESTPE.2020.3015128.

[4] M. B. Shadmand, S. Jain, and R. S. Balog, "Autotuning Technique for the Cost Function Weight Factors in Model Predictive Control for Power Electronic Interfaces," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 7, no. 2, pp. 1408-1420, 2019, doi: 10.1109/JESTPE.2018.2849738.

[5] S. Jain, M. B. Shadmand, and R. S. Balog, "Decoupled Active and Reactive Power Predictive Control for PV Applications Using a Grid-Tied Quasi-Z-Source Inverter," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 6, no. 4, pp. 1769-1782, 2018, doi: 10.1109/JESTPE.2018.2823904.

[6] M. Hosseinzadehtaher, A. Khan, M. Easley, M. B. Shadmand, and P. Fajri, "Self-Healing Predictive Control of Battery System in Naval Power System With Pulsed Power Loads," *IEEE Transactions on Energy Conversion,* vol. 36, no. 2, pp. 1056-1069, 2021, doi: 10.1109/TEC.2020.3014294.

[7] S. Vazquez *et al.*, "Model Predictive Control: A Review of Its Applications in Power Electronics," *Industrial Electronics Magazine, IEEE,* vol. 8, pp. 16-31, 2014.

[8] H. Althuwaini and M. B. Shadmand, "Battery Sources Power Balancing in a Cascaded Multilevel Inverter via an Optimal Moving Horizon Predictive Control," in *IECON 2021 The 47th Annual Conference of the IEEE Industrial Electronics Society*, 2021.

[9] K. Lee, B. Park, R. Kim, and D. Hyun, "Robust Predictive Current Controller Based on a Disturbance Estimator in a Three-Phase Grid-Connected Inverter," *IEEE Transactions on Power Electronics,* vol. 27, no. 1, pp. 276-283, 2012, doi: 10.1109/TPEL.2011.2157706.

[10] O. H. Abu-Rub, A. Y. Fard, M. F. Umar, M. Hosseinzadehtaher, and M. B. Shadmands, "Towards Intelligent Power Electronics-Dominated Grid via Machine Learning Techniques," *IEEE Power Electronics Magazine,* vol. 8, no. 1, pp. 28-38, 2021, doi: 10.1109/MPEL.2020.3047506.

[11] M. Easley, A. Y. Fard, F. Fateh, M. B. Shadmand, and H. Abu-Rub, "Auto-tuned Model Parameters in Predictive Control of Power Electronics Converters," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 29 Sept.-3 Oct. 2019 2019, pp. 3703-3709, doi: 10.1109/ECCE.2019.8912881.

[12] Y. Li, Y. Li, and Q. Wang, "Robust Predictive Current Control With Parallel Compensation Terms Against Multi-Parameter Mismatches for PMSMs," *IEEE Transactions on Energy Conversion,* vol. 35, no. 4, pp. 2222-2230, 2020, doi: 10.1109/TEC.2020.3002274.

[13] H. Liu and S. Li, "Speed Control for PMSM Servo System Using Predictive Functional Control and Extended State Observer," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 2, pp. 1171-1183, 2012, doi: 10.1109/TIE.2011.2162217.

[14] H. A. Young, M. A. Perez, and J. Rodriguez, "Analysis of Finite-Control-Set Model Predictive Current Control With Model Parameter Mismatch in a Three-Phase Inverter," *IEEE Transactions on Industrial Electronics,* vol. 63, no. 5, pp. 3100-3107, 2016, doi: 10.1109/TIE.2016.2515072.

[15] J. Rodríguez, R. Heydari, Z. Rafiee, H. A. Young, F. Flores-Bahamonde, and M. Shahparasti, "Model-Free Predictive Current Control of a Voltage Source Inverter," *IEEE Access,* vol. 8, pp. 211104-211114, 2020, doi: 10.1109/ACCESS.2020.3039050.

[16] H. Ren, W. Song, and Z. Ruan, "Model Predictive Control of PMSM Considering Online Parameter Identification," in *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, 7-9 Sept. 2019 2019, pp. 1307-1311, doi: 10.1109/CIEEC47146.2019.CIEEC-2019478.

[17] M. Easley, M. Baker, A. Khan, M. B. Shadmand, and H. Abu-Rub, "Self-healing Model Predictive Controlled Cascaded Multilevel Inverter," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 29 Sept.-3 Oct. 2019 2019, pp. 239-244, doi: 10.1109/ECCE.2019.8913011.

[18] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulò, "Deep Neural Decision Forests," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 7-13 Dec. 2015 2015, pp. 1467-1475, doi: 10.1109/ICCV.2015.172.

[19] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," *arXiv preprint arXiv:2003.05689,* 2020.