

Using Machine Learning Technology to Online Predict the Maximum Common Mode Current of Three-phase Motor Drive Inverter

Ximu Zhang
Min H. Kao Department of
Electrical Engineering and
Computer Science
The University of Tennessee-
Knoxville
Knoxville, TN, USA
xzhan134@vols.utk.edu

Yang Huang
Min H. Kao Department of
Electrical Engineering and
Computer Science
The University of Tennessee-
Knoxville
Knoxville, TN, USA
yhuang65@vols.utk.edu

Jared Walden
Min H. Kao Department of
Electrical Engineering and
Computer Science
The University of Tennessee-
Knoxville
Knoxville, TN, USA
jwalde12@vols.utk.edu

Hua Bai
Min H. Kao Department of
Electrical Engineering and
Computer Science
The University of Tennessee-
Knoxville
Knoxville, TN, USA
hbai2@utk.edu

Fanning Jin
Mercedes Benz North America
Detroit, MI, USA
fanning.jin@daimler.com

Xiaodong Shi
Mercedes Benz North America
Detroit, MI, USA
xiaodong.shi@daimler.com

Bing Cheng
Mercedes Benz North America
Detroit, MI, USA
bing.cheng@daimler.com

Abstract— To reduce the common-mode voltage (CMV) in the PWM-based motor drive system, many CMV reduction methods have been proposed. However, the performance of such methods has limitations such as only being implemented on particular operating conditions with fixed switching frequency or PWM patterns and relying on the simulation or experimental data. This paper explores machine-learning-based methods to actively evaluate the CM performance. Machine learning methods are employed to actively analyze three popular PWMs (SVPWM, AZSPWM, and DPWMMin) on-chip. In this way, we can online determine the best PWM pattern and switching frequency with a minimum requirement of computation resources based on the torque and speed command.

Keywords—CMV reduction, motor drive system, machine learning

I. INTRODUCTION

A three-phase voltage source inverter (VSI), as shown in Fig. 1, is commonly used in AC motor drive systems [1]. As a result of common-mode voltage (CMV), the common-mode current (CMI) flows through the motor case to the capacitor middle point. This will cause insulation failure, greatly reducing the motor lifespan, and cause significant EMI problems. To reduce the CMV, and ultimately reduce the CMI, some PWM methods have been proposed, such as active zero state PWM (AZSPWM) and discontinuous PWM clamped to the negative DC bus (DPWMMin). As shown in Fig. 2, the common idea for these PWMs is to reduce or avoid the usage of zero-state vectors. Compared with space vector PWM (SVPWM), the peak value of the CMV has been shaved by AZSPWM and DPWMMin.

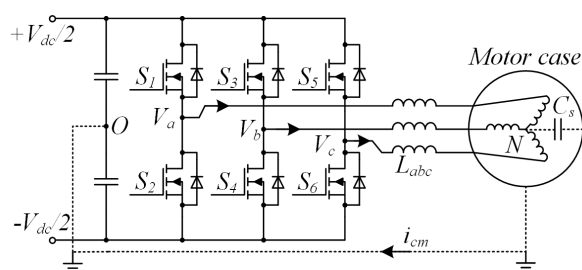


Fig. 1 Three-phase two-level VSI motor drive system

To quantify the actual CMI reduction performance, traditional approaches count on simulation and experiments to collect the data at specific operating points, which lacks analytical models and tools. Reference [2] proposed a double-Fourier integral (DFI) based analytical model to characterize the CM performance of various PWMs. However, such a DFI model is significantly time-consuming, taking tens of minutes to finish the computation on the ARM Cortex-A9 processor on Xilinx Zynq-7000 Zedboard. This makes it infeasible to do an online calculation of CMI. Although using a lookup table (LUT) is a good alternative, it may require a lot of memory space, especially the table could be multi-dimensional. Another method is to use a machine learning (ML) algorithm to learn the mapping between the input parameters and the target. In this case, only the trained model parameters need to be stored in the memory. The required memory space depends on the complexity of the model (number of parameters) rather than the data itself and the acceptance of runtime can also be guaranteed, which makes the online estimation of the CMI possible, thereby selecting PWM patterns based on CMI is also possible.

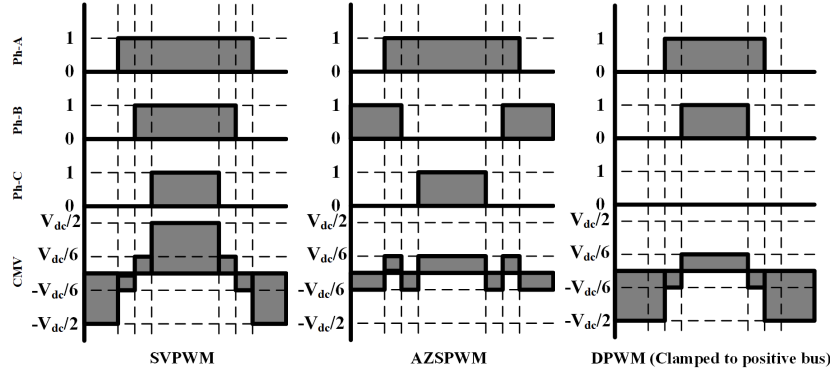


Fig. 2 Various pulse patterns and corresponding CMV distribution (left) SVPWM (middle) AZSPWM (right) DPWMMin

II. LOOKUP TABLE VS MACHINE LEARNING FOR MAXIMUM CMI ESTIMATION

A. Lookup-table-based Method

A lookup table estimates the new data that is not covered in the table by using some interpolation methods. For a 3-D lookup table, the bilinear interpolation [3] is the simplest and easiest one to be implemented in the embedded systems/microprocessors. Assume there are 4 data (f_1, M_1, I_{11}) , (f_1, M_2, I_{12}) , (f_2, M_1, I_{21}) and (f_2, M_2, I_{22}) in the lookup table where f is the inverter switching frequency, M is the modulation index and I is the peak CMI ($f_1 < f_2, M_1 < M_2$). Any data on the surface formed by these four data points can be estimated using bilinear interpolation. Assume the unknown data is (f, M, I) where f and M are given. The estimated peak CMI at (f, M) can be obtained by the following equation.

$$I = \frac{M_2 - M}{M_2 - M_1} \times \left(\frac{f_2 - f}{f_2 - f_1} \times I_{11} + \frac{f - f_1}{f_2 - f_1} \times I_{21} \right) + \frac{M - M_1}{M_2 - M_1} \times \left(\frac{f_2 - f}{f_2 - f_1} \times I_{12} + \frac{f - f_1}{f_2 - f_1} \times I_{22} \right) \quad (1)$$

B. Machine-learning-based Method

Estimating the maximum CMI is a regression problem. Various ML algorithms aiming to solve such problems have already been proposed and applied. Linear regression is the simplest regression algorithm where each feature has a linear relationship with the target, as shown in the following equation.

$$f(\mathbf{x}) = \boldsymbol{\omega}^T \mathbf{x} + \mathbf{b} \quad (2)$$

$$J = \frac{1}{n} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 \quad (3)$$

where $\boldsymbol{\omega}$, \mathbf{x} and \mathbf{b} are vectors, and the loss function is used to update the set $(\boldsymbol{\omega}, \mathbf{b})$ that gives the minimum error given the dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $x_i \in R^d, y_i \in R^l$.

There are other improved regression models such as Ridge Regression and Lasso Regression which aim to address the overfitting issue and learn a sparse model. However, these are all linear regression models which can guarantee good prediction performance in problems where the features have high linearity with the target. For the non-linear regression problems, without any feature engineering, those linear models are not good candidates. Thus, other models need to be raised to

solve non-linear regression problems. Neural networks (NN) consist of series of hidden layers containing multiple neurons. Each neuron processes equation (2) and passes the result to a non-linear activation function. Because of activation functions in the neural networks (NN), such models yield better performance and are able to learn any functions in theory [4].

III. MAXIMUM COMMON MODE CURRENT ESTIMATION – DATASET AND MODEL STRUCTURE

A. Dataset Collection

As mentioned in Section I and Section II, the DFI takes a long time to execute on the ARM processor, making it impossible to online estimate the CMI. Here either lookup table or NN can replace DFI and directly obtain the maximum CMI at any operating point. Such maximum CMI then can be compared with vehicle standards such as CISPR 25 to online examine if any EMI requirement has been violated.

To form the lookup table or train the NN, the dataset needs to be collected. In this paper, we assume the motor inverter has no CM choke. The DC bus voltage and fundamental frequency are constant which are 400 V and 100 Hz, respectively. In LUT, the inputs are modulation index (MI) and switching frequency (fs), and the output is the estimated peak CMI. In NN, modulation index and switching frequency are the features and the peak CMI is the target. Both training dataset and test dataset are collected. The training dataset is used to form a lookup table or train a neural network model and the test dataset is used to validate models.

The modulation index in the training dataset varies from 0.1 to 0.9 with a step size of 0.05, while the switching frequency is from 10 kHz to 40 kHz with a step size of 250 Hz. The test dataset is collected with less data where the modulation index is from 0.13 to 0.87 with the step size of 0.1, and the switching frequency is from 14 kHz to 38 kHz with the step size of 5 kHz. Three PWMs are used, therefore the peak CMI values under each PWM are collected for the training dataset and test dataset. In summary, there are 2057 samples in total in the training dataset and 40 samples in the test dataset, where each sample contains fs, MI, and three peak CMI values for SVM, AZPWM, and DPWMMin. Fig. 3 shows the distribution of the data in the training dataset and test dataset in 2D point of view (fs and MI). The surface plot of the training dataset is shown in Fig. 5.

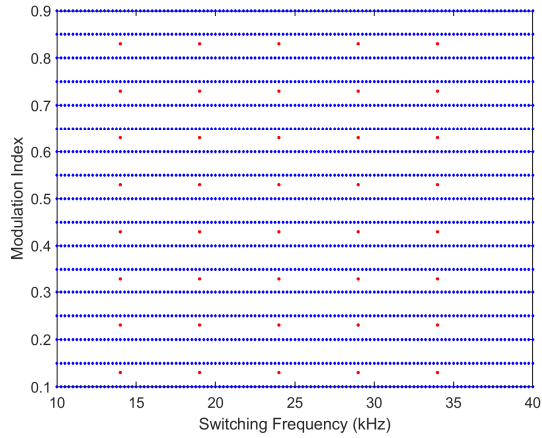


Fig. 3 Data distribution in 2D point of view (blue points are training data; red points are test data).

According to these figures, it obviously reveals the non-linearity between the features (MI, f_s) and the target (peak CMI).

B. Model Structure

To implement and test the LUT model, the whole training dataset is stored in the memory and the test dataset is used for evaluation only. The LUT model structure is so simple that it will not be covered here. To implement and test the NN model, the training dataset is used to train a model in a PC and only the trained parameters (weights and biases) are stored in the memory. The structure of the NN model is shown in Table 1 and Fig. 4.

Table 1 Parameters of the neural network

Layers	# of neurons	# of weights	# of biases	Activation function
Input Layer	2			
Hidden Layer 1	4	8	4	Tahn
Hidden Layer 2	8	32	8	Relu
Hidden Layer 3	16	128	16	Relu
Output Layer	1	16	1	Liner

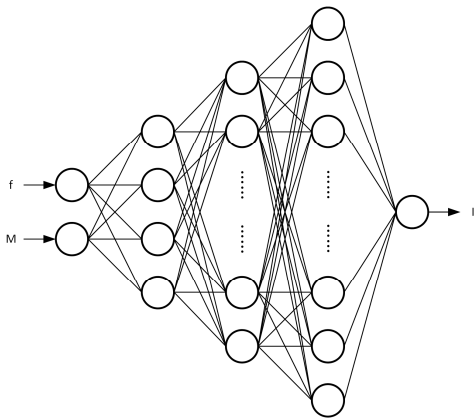
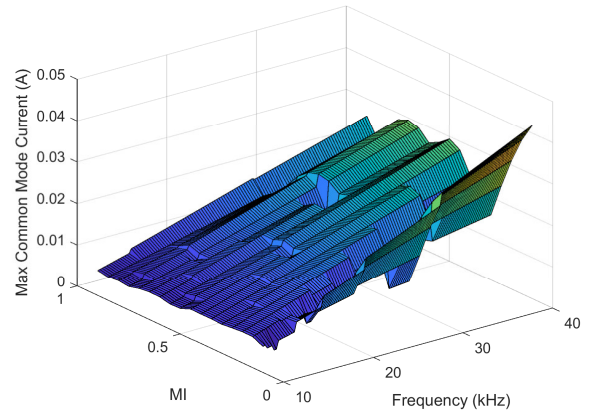
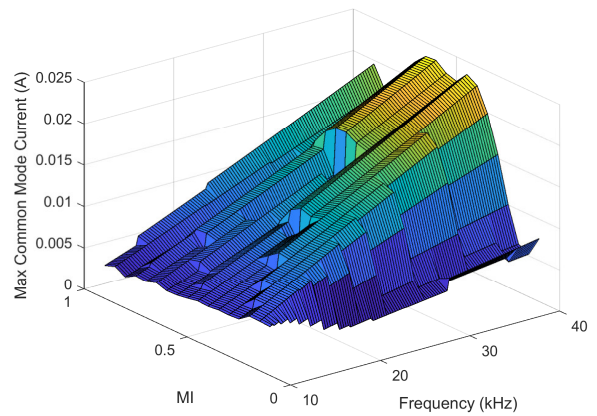


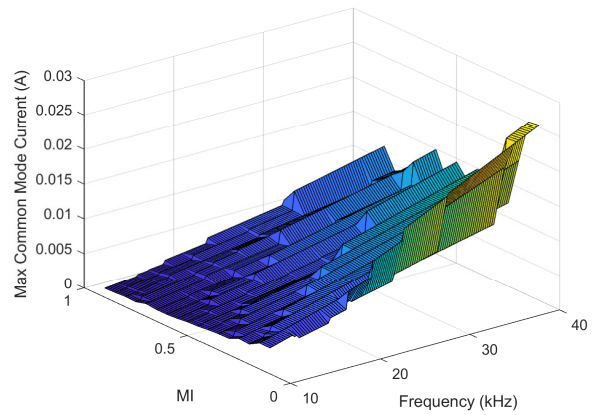
Fig. 4 Structure of the proposed neural network (f : switching frequency; M : modulation index; I : peak CMI).



(a)



(b)



(c)

Fig. 5 Surface plot of peak CMI with different f_s and MI for: (a) SVM (b) AZSPWM (c) DPWMMIn.

IV. IMPLEMENTATION AND EVALUATION

In the training process of the NN, the training dataset is split into the training set and validation set based on the ratio of 4:1. The training process will stop when the validation performance

has increased more than the maximum failure times, which is 10 in our training algorithm. Levenberg-Marquardt backpropagation is used as the training function. The pseudocode of the implementation of the lookup table and NN are given in algorithm 1 and algorithm 2, respectively, as shown below.

Algorithm 1 Implementation of the lookup table.

- Require:** T , the lookup table. f , switching frequency. M , modulation index.
- 1: obtain f and M
 - 2: if the pair (f, M) exists in T then
 - 3: retrieve the I from T
 - 4: else
 - 5: find f_1 and f_2 that is closest to f ($f_1 < f \leq f_2$ or $f_1 \leq f < f_2$)
 - 6: find M_1 and M_2 that is closest to M ($M_1 < M \leq M_2$ or $M_1 \leq M < M_2$)
 - 7: retrieve the data from T : (f_1, M_1, I_{11}) , (f_1, M_2, I_{12}) , (f_2, M_1, I_{21}) and (f_2, M_2, I_{22})
 - 8: use bilinear interpolation (3) to calculate I
 - 9: end if
-

Algorithm 2 Implementation of neural network.

- Require:** ω , trained weights. b , trained biases. f , switching frequency. M , modulation index. n_l , # of layers.
- 1: obtain f and M
 - 2: normalization: $(f, M) \rightarrow (f_{norm}, M_{norm})$
 - 3: $x_0 = [f_{norm}, M_{norm}]^T$
 - 4: for $l = 0, \dots, n_l - 1$ do
 - 5: $x_{l+1} = \text{activate}(\omega_l^T x_l + b)$
 - 6: end if
 - 7: $I = x_{n_l}$
-

The prediction performance of LUT and NN is evaluated using the test dataset (containing 40 samples), and the results are shown in Fig. 6. A dot symbol "." denotes a real peak CMI value. A plus symbol "+" denotes an estimated peak CMI value using the lookup table. A circle symbol "o" denotes an estimated peak CMI value using NN. Fig. 7 plots the prediction errors and mean errors of LUT and NN models evaluated on the test dataset. Absolute errors are used here. Table 2 shows the summary of the models and time/space expense using ARM Cortex-A9 processor on Xilinx Zynq-7000 Zedboard. The program runtime is measured using a global timer on the board and the occupied space is calculated based on the data type and number of the data stored in the memory.

Based on the results, the lookup table has shorter runtime (at microsecond level), but it requires much larger space to store data compared with NN. In other words, the NN sacrifices time to save space, which has millisecond-level runtime but just needs 30 times less memory. The difference in runtime between the LUT model and the NN model is reasonable. Lookup is quick! And NN model involves some matrix computations which slow down the algorithm execution time.

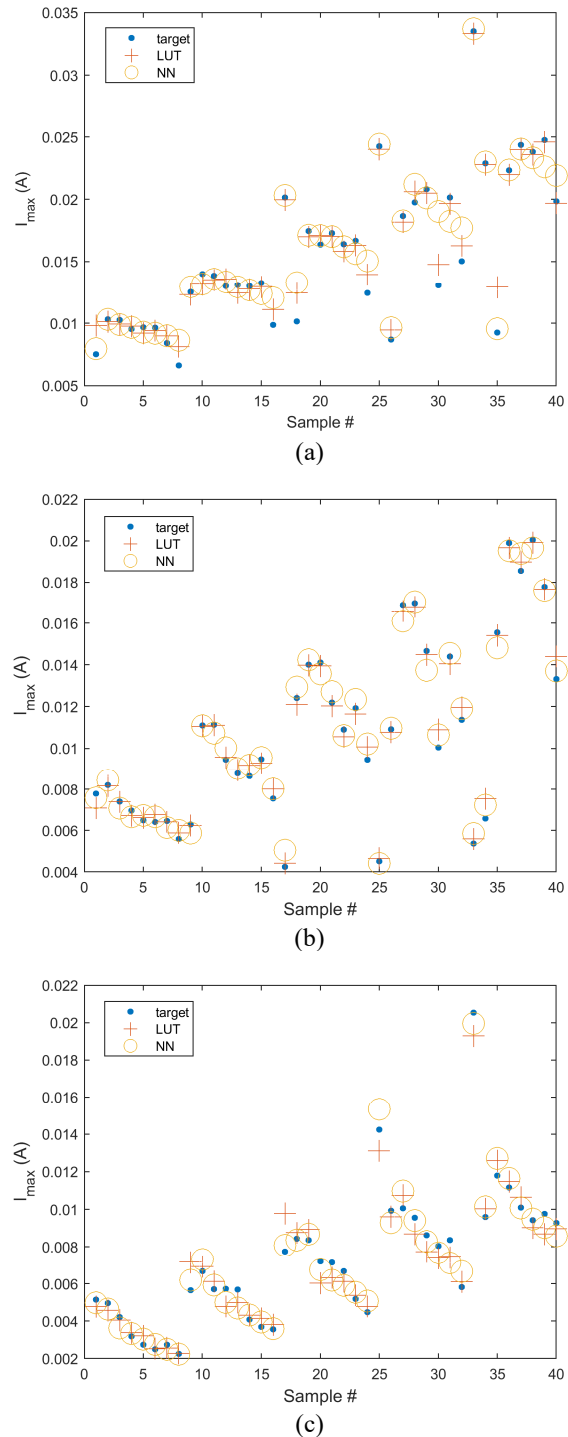


Fig. 6 Evaluation of the lookup table and neural network for: (a) SVM, (b) AZSPWM, (c) DPWMMin.

The prediction results of LUTs on SVM and AZSPWM are slightly better than NNs while on DPWMMin, NN models are better than LUTs. Some prediction errors could reach as high as 20% but in most test cases, the errors are distributed under 15%.

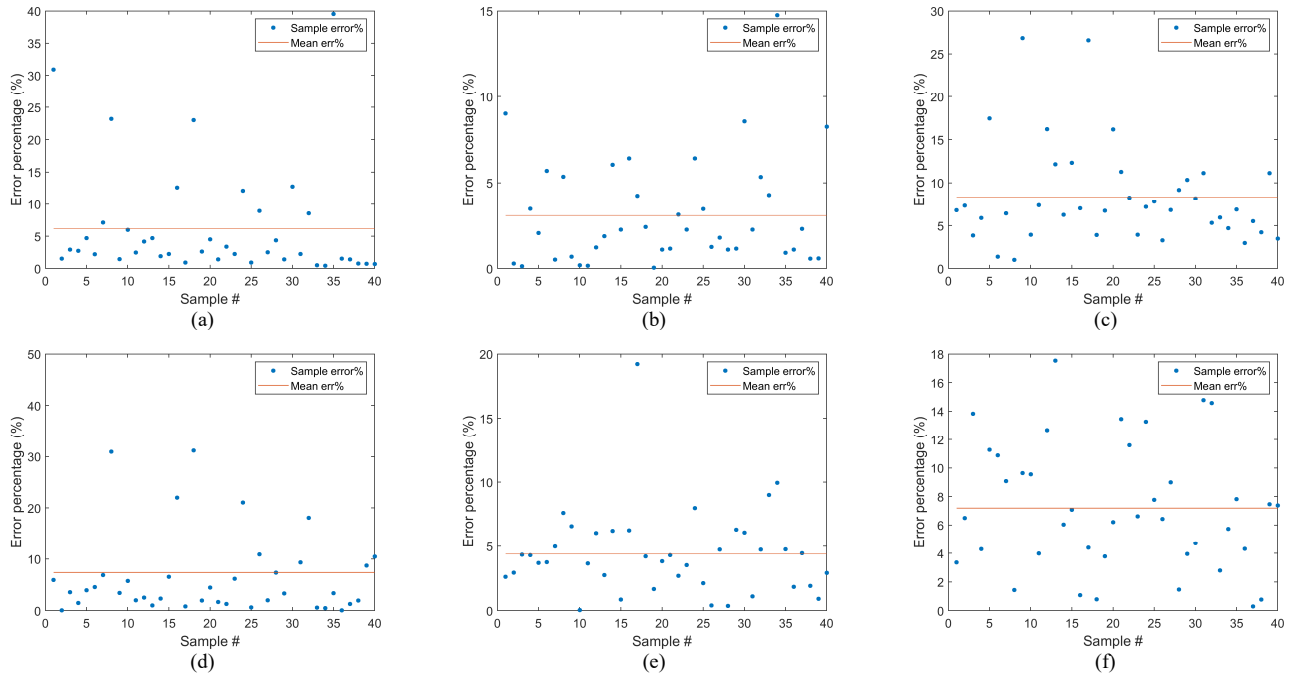


Fig. 7 Model evaluation error plots on test dataset (Row 1: LUT models; Row 2: NN models) (Col 1: SVM; Col 2: AZSPWM; Col 3: DPWMMin).

In the data collected, the common-mode currents, at the milliamper level, are very small. Such errors will not affect the results a lot. It should be also addressed that the neural network can further be modified and improved to get better prediction performance. In real applications in EV, the runtime of the neural network has already been acceptable for the on-chip real-time operation. Note that this computation process can also be speeded up by using some hardware acceleration techniques or choosing other microprocessors with higher performance.

Table 2 Summary of models and time/space expense

Model	Runtime (ms)	Data Stored	Occupied Space (KB)	Prediction Error MSE
SVM LUT	0.00061725	6171 float32	24.684	1.0211e-06
SVM NN	0.17469	213 float32	0.852	2.2513e-06
AZSPWM LUT	0.0006393	6171 float32	24.684	1.5002e-07
AZSPWM NN	0.17407	213 float32	0.852	2.2194e-07
DPWMMin LUT	0.00063765	6171 float32	24.684	5.2204e-07
DPWMMin NN	0.17461	213 float32	0.852	3.3076e-07

Online estimation of peak CMI has various potential applications. Once the model is deployed in the embedded system, we can then online decide the best switching frequency and PWM to comply with the CMI standard and process other optimization based on CMI.

V. EXPERIMENT RESULTS

LUT and NN models for three PWM patterns are built, trained, and evaluated. In this section, all the models are

deployed in the ARM Cortex-A9 processor and verified through experiments.

Fig. 8 shows the test benchmark. The motor drive system consists of a SiC inverter, a two-pole three-phase PMSM, a Xilinx Zynq-7000 Zedboard (ARM + FPGA), and a load. Three different PWM patterns mentioned in this paper are generated by the Zedboard to drive the motor. The EMI receiver is used to measure the CMI and generate the spectrum. In this paper, we only focus on the spectrum from 150 kHz to 300 kHz, and the predicted peak CMI is also within this region.

Fig. 9 shows the measurement results at switching frequency: 10 kHz and modulation index: 0.31. Within the frequency range from 150 kHz to 300 kHz, the maximum common mode currents are labeled in the figures. The maximum

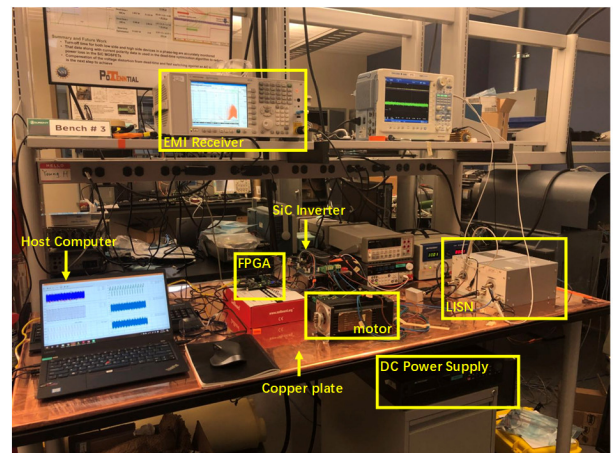


Fig. 8 Experiment test benchmark.

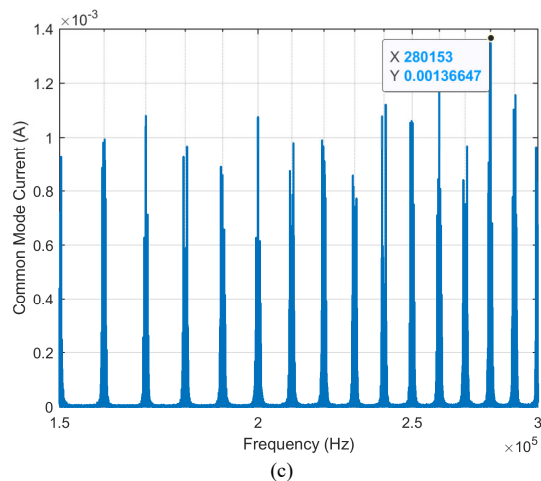
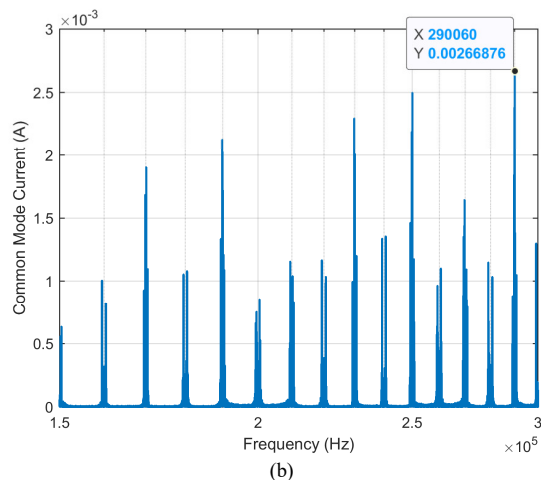
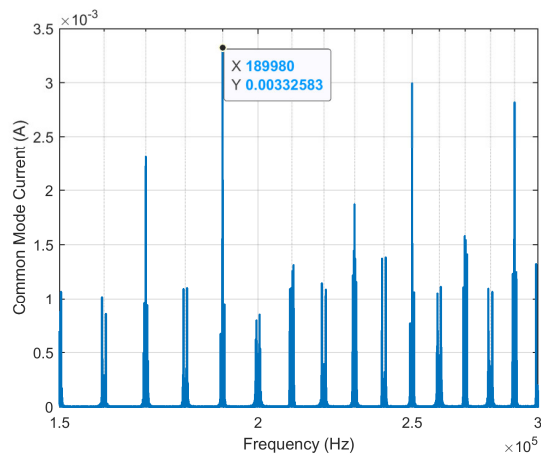


Fig. 9 Measured CMI spectrum (150 kHz – 300 kHz) with PWM: (a) SVM; (b) AZSPWM; (c) DPWMMin.

common mode currents with SVM, AZSPWM and DPWMMin are 3.32583 mA, 2.66876 mA and 1.36647 mA, respectively.

The predicted peak common mode currents generated by LUT and NN models are compared with the measurement results in Table 3 and Table 4. Absolute error is used here. The

absolute error percentages are around 10% which match the model performance on the test dataset.

Table 3 Predicted peak CMI by LUT model

PWM	Measurement (A)	LUT Prediction (A)	Error (%)
SVM	0.00332583	0.00344904	3.7047
AZSPWM	0.00266876	0.00245095	8.1616
DPWMMin	0.00136647	0.00116997	14.3798

Table 4 Predicted peak CMI by NN model

PWM	Measurement (A)	NN Prediction (A)	Error (%)
SVM	0.00332583	0.00346765	4.2641
AZSPWM	0.00266876	0.00244687	8.3145
DPWMMin	0.00136647	0.00125049	8.4876

VI. CONCLUSION AND FUTURE WORK

This paper proposes a method to solve the computation issue in the CMI estimation through a machine learning perspective compared with a lookup table, aiming to save the computation resource consumed by the DFI. The lookup table and neural network give similar performance in predicting maximum CMI. The neural network model saves more space than the lookup table does. In applications where lookup tables need to store lots of data, neural networks could be potential alternatives to replace lookup tables. In addition, at the current stage, DC bus voltage and fundamental frequency are constants. However, in electric vehicles, such two parameters are variable, so that using the lookup table is not an ideal choice because it will take too much space to store data for a different combination of modulation index, switching frequency, DC bus voltage, and fundamental frequency. Several large lookup tables may be generated in order to predict the peak CMI at different working operation points. In such case, a neural network could use one model to handle, treating those variables as input features and peak CMI as output target. Due to the limitation of data collected, neural network models in this paper still have room to be improved. After that, such ML-based CMI estimation module can be deployed on ARM to online decide PWM patterns, which will be covered in further research.

ACKNOWLEDGMENT

This work was funded by Mercedes-Benz R&D North America. The experimental validation made use of the Engineering Research Center Shared Facilities supported by the Engineering Research Center Program of the National Science Foundation and DOE and the CURENT Industry Partnership Program.

REFERENCES

- [1] Harahap, Charles Ronald, and Tsuyoshi Hanamoto. "Speed control of permanent magnet synchronous motor using FPGA for high frequency SiC MOSFET inverter." *Journal of Engineering Science and Technology (JESTEC)* 9 (2014): 11-20.
- [2] Huang, Yang, et al. "Analytical Characterization of CM and DM Performance of Three-Phase Voltage-Source Inverters Under Various PWM Patterns." *IEEE Transactions on Power Electronics* 36.4 (2020): 4091-4104.

- [3] L. Jing, S. Xiong and W. Shihong, "An Improved Bilinear Interpolation Algorithm of Converting Standard-Definition Television Images to High-Definition Television Images," 2009 WASE International Conference on Information Engineering, Taiyuan, Chanxi, 2009, pp. 441-444, doi: 10.1109/ICIE.2009.251.
- [4] H. Lin and S. Jegelka, "ResNet with one-neuron hidden layers is a universal approximator," in Proc. Adv. Neural Inf. Process. Syst., Montreal, CA, Dec. 3-8, 2018, pp. 6169-6178.